

# L<sup>A</sup>T<sub>E</sub>X 文档中文参考文献的 BibLaTeX 解决方案

胡振震<sup>1</sup>

( 1. UNIT CHINA 100000 )

**摘 要** 针对中文 L<sup>A</sup>T<sub>E</sub>X 环境的参考文献生成问题, 以入门教程为出发点, 介绍了文档源代码、bib 数据库、编译方法、文献表生成、格式修改、多语言、筛选和排序等大量细节问题, 为国内用户提供了一套基于 BibLaTeX 宏包的完整参考文献解决方案。介绍了 biblatex-gb7714-2015 样式包和一个样式定制示例的设计思路和实现过程, 揭示了基于 BibLaTeX 标准样式定制样式的一般原理和开发方法, 为 BibLaTeX 参考文献样式定制提供了完整示范。最后针对使用传统方法的用户需求, 总结了基于 thebibliography 环境和基于 bibtex 程序的参考文献生成方法。

**关键词** L<sup>A</sup>T<sub>E</sub>X 参考文献 BibLaTeX 宏包

**中图分类号** (V211.751)

**文献标志码** A

## 0 引言

参考文献是科技文档写作中的重要内容, 在写作中常碰到这样的问题, 比如: 怎么生成满足格式要求的参考文献表? 正文中的文献引用如何能够产生满足格式要求的标注标签? 怎么生成分章文献表和统一的全书文献表? 等等。这些问题尽管不会成为困扰, 但可能牵扯文档作者不少精力。为使 L<sup>A</sup>T<sub>E</sub>X 文档作者能更便捷地生成参考文献, 有必要对相关问题进行总结梳理, 并给出简单解决方案, 以便随时参考和应用。

对于一般需求而言, 利用传统的 L<sup>A</sup>T<sub>E</sub>X 参考文献生成方法<sup>[1-3]</sup> 配合相关的 bst 样式文件基本能够满足要求。但如果要实现一些特殊格式或要求, 就需要应用一些辅助宏包, 比如 chapterbib(用于分章文献表)<sup>[4]</sup>, multibib(用于单文档内生成多文献表)<sup>[5]</sup>, natbib(用于标注并提供著录样式)<sup>[6]</sup>, mcitesplus(用于标注)<sup>[7]</sup>, inlinebib(用于文献表置于脚注中)<sup>[8]</sup> 等。这些工具从多个方面提供了需求功能, 但多个宏包同时使用可能存在不兼容等问题。另外, 由于使用 bibtex/bibtex8 程序本身的原因, 非 utf-8 编码的 bib 文件无法使用中文的引用关键字, 中文文献排序往往也需要附加很多特殊处理, 甚至当参考文献量足够大时可能出现内存不足的情况。

而 biblatex 宏包<sup>[9]</sup> 旨在提供了一套完整的方案, 既能解决传统方法的一些局限, 也能避免大量使用辅助宏包, 更能提供很多新功能, 比如动态数据修改等。biblatex 的强大和完善使它成为 L<sup>A</sup>T<sub>E</sub>X 参考文献生成的一个绝佳选择。由于使用习惯及存量 bst 广泛应用, 传统方法无法完全取代, 但由于 biblatex 的大量优点, 基于 biblatex 的方法呈逐渐流行态势, 目前 CTAN 上面已经有包括 APA、MLA、science、ieee、nature、iso690、gb7714-2015 等数十种第三方样式可以选用, 结合 biblatex 的强大定制功能, 利用 biblatex 一个宏包完全可以简单而直接满足 L<sup>A</sup>T<sub>E</sub>X 文档参考文献相关所有可能需求。

The files have version number v1.0h, last revised 2023 年 4 月 9 日.

第一作者简介: HU ZHENZHEN, A L<sup>A</sup>T<sub>E</sub>X LOVER, Email: hzzmail@163.com。

然而正是由于 biblatex 的完整性, biblatex 宏包相当的庞大, 其说明文档 (中译版) 长达 300 页。对于初学者或者无深度需求的使用者来说, biblatex 手册就像一个技术说明书, 详实的介绍宏包功能和接口, 但却没有提供一个像 lshort 这样的简单入门教程。而类似 biblatex-gb7714-2015 这样的参考文献 biblatex 样式包<sup>[10]</sup>, 虽然也介绍了一些 biblatex 的使用方法但却并不详细。

总的来说, 对于 biblatex 参考文献生成, 目前还没有较为系统的入门手册<sup>1</sup>。本文正是出于这样的现状考虑, 希望从初级用户的角度, 面向应用, 以示范的形式, 给出一个简易但全面的使用说明书, 为用户入门提供帮助。

需要提示的是: 本文所有的示例和测试都基于 texlive 发行版, 2014/2015/2016/2017/2018 及以后版本均可, 文档均采用 UTF-8 编码, 编译使用 xelatex+biber, texlive 的安装和使用可以参考 texlive 安装手册<sup>[11]</sup> 以及 [ctex 论坛](#) [ctex 论坛](#) 或 [latex sudio](#) 上分享的各种方法。

## 1 为什么使用 biblatex

### 1.1 L<sup>A</sup>T<sub>E</sub>X 参考文献基本原理

L<sup>A</sup>T<sub>E</sub>X 中参考文献生成主要借助于 \thebibliography 环境。这个环境可以手工在 tex 源文档中输入, 也可以利用外部程序生成, 如 bibtex 程序。

首先我们看一下手工写 \thebibliography 环境的方法。典型的 \thebibliography 环境如例 1 所示。在该环境中每一条文献都用一条 \bibitem 命令组织, 其第一个可选参数信息可以用于生成引用标注标签 (如例中的 {Baker et al.(1995)Baker and Jackson}), 其后的必选参数则是该条参考文献的引用关键词 (如例中的 {baker}), 最后跟着的是用于生成参考文献表的著录内容。

示例 1 典型的 thebibliography 环境

```
1 \begin{thebibliography}{2}
2 \bibitem[{Baker et al.(1995)Baker and Jackson}]{baker}BAKER-S-K, JACKSON-M-E. \newblock The future of
   resource sharing\allowbreak[M]. \newblock New York: The Haworth Press, 1995.
3
4 \bibitem[{Chernik(1982)}]{chernik}CHERNIK-B-E. \newblock Introduction to library services for library
   technicians\allowbreak[M]. \newblock Littleton, Colo.: Libraries Unlimited, Inc., 1982.
5 \end{thebibliography}
```

代码

在文档编译过程中, 第一遍编译时, 基于 \bibitem 命令的宏替换, T<sub>E</sub>X 编译程序 (如 pdf<sub>l</sub>atex、xelatex、lua<sub>l</sub>atex 等) 将在 aux 文件中写入对应每条 \bibitem 命令的标签生成命令 \bibcite。第二遍编译时, T<sub>E</sub>X 编译程序会读入 aux 文件并根据 \bibcite 命令生成对应引用关键词的标签, 编译过程中当遇到 \cite 命令 (该命令的参数中给出了当前引用文献的引用关键词), 则调用对应引用关键词的标签, 以生成引用的标注标签。

基于外部工具 bibtex 的参考文献生成方法, 本质上是类似的, 也需要 \thebibliography 环境。只是利用了外部程序 bibtex 生成该环境, 而不是在 tex 源文档中手工写入。在 tex 源文档中, 利用命令 \bibliographystyle 命令指定 bibtex 程序需要使用的 bst 文件, 利用命令 \bibliography 命令指定 bibtex 程序需要处理的 bib 文件。在源文档第一步编译中, T<sub>E</sub>X 编译程序将这两个文件信息写入到 aux 文件中。接着运行 bibtex 程序, 该程序会读取 aux

<sup>1</sup> 准确的说是中文入门手册, 因为:

biblatex 3.12 版以后, 宏包作者提供了一个简单的 [入门教程](#) (见 [pdf 文档](#))

另外 Paul Stanley 也提供了一个 [入门教程](#) (见 [pdf 文档](#)), 该教程对基本概念介绍相当清晰, 感兴趣的读者可以细看。

文件中的两个指定文件, 并处理生成一个 bbl 文件, 该文件中包含 `\thebibliography` 环境。源文档第二遍编译时, T<sub>E</sub>X 编译程序读取 bbl 文件, 并利用 `\bibitem` 命令将标签生成命令 `\bibcite` 写入到 aux 文件中。源文档第三遍编译时, T<sub>E</sub>X 编译程序再次读取 aux 文件并依据 `\cite` 命令将标签输出。

其中参考文献的著录格式由 `\thebibliography` 环境的 `\bibitem` 命令必选参数后跟着的文献内容决定, 比如例 1 中的第一条参考文献的格式是 {baker} 之后的内容决定的, 因此可以采用手动输入方式来实现需要的格式, 或者由满足要求的 bst 样式文件来生成。而引用文献的标注标签格式则由 `\bibcite` 命令的定义及其参数决定。该命令的定义在 latex 中有原始的定义, 一些宏包比如 natbib, jurabib 等会对其重新定义以生成满足某种要求的格式。但如果该命令的输入参数中如果没有足够的信息, 那么再多的格式要求也会巧妇难为无米之炊。因此要实现一些特殊的格式就需要为其提供足够的文献信息。而这一信息又来源于 `\bibitem` 命令的写入 aux 操作, `\bibitem` 命令通常会将其后跟着的 [] 中的可选参数内容当做 `\bibcite` 命令的参数, 因此引用的标注标签生成的关键之一是在 `\bibitem` 后面的 [] 内写入足够文献信息, 同样这也可以手工写入, 也可以利用样式文件 bst 生成。对于 natbib 宏包来说, 通常需要在 `\bibitem` 命令后的 [] 中写入三个信息: 作者的缩写信息、由 () 包围的年份信息、以及作者的详细信息, 其中 () 作为定界符来读取这三个信息。因此利用 natbib 宏包生成标注标签时, 通常要在 `\bibitem` 后面的 [] 内做文章, 无论是通过手工写的方法还是通过 bst 样式文件生成。jurabib 宏包也有类似情况, 只是其解析方式更为复杂, 需要写入更多的信息, 以便在标签生成时生成更多的内容, 比如文献的题名等。

总结起来, L<sup>A</sup>T<sub>E</sub>X 参考文献生成的基本方法是利用 `\thebibliography` 环境写参考文献的著录格式, 并利用 `\bibitem` 命令将其可选参数信息输出, 作为引用文献标注标签生成的基础在正文中生成参考文献引用标签。如果要采用非手工方式生成该环境, 那么就要利用外部程序 bibtex 和参考文献样式 bst 文件以及参考文献的数据源 bib 文件。其中 bib 文件是以 BibTeX 格式统一规范的文本文件, 该格式简单且通用, 是默认的参考文献数据源标准, bib 文件作为文本文件也非常容易编辑, 关于该格式在第 2.2 节还有详细介绍。而 bst 则是与 bibtex 程序配合使用的样式文件, 以一种能被 bibtex 程序识别的语法实现, 该语法与 TeX 基本语法迥异, 因此一般用户学习起来不太容易, 因此要设计某种样式的 bst 文件也相对困难。而 bibtex 程序则是用于处理 bib 文件生成 bbl 文件的程序, 通常由 TeX 发行版 (如 Texlive、MacTeX 等) 附带。

## 1.2 BibLaTeX 参考文献生成的基本原理

基于 BibLaTeX 的参考文献生成方法, 从根子上就不再利用 `\thebibliography` 环境, 但操作方式有点类似于传统的基于 bibtex 程序的方法, 相似点在于同样利用外部程序处理参考文献数据源, 也同样需要利用辅助文件。BibLaTeX 的设计出发点就是要达成两个主要目的, 一是要避免传统参考文献生成方法过度的读写辅助文件, 二是要利用 TeX/LaTeX 命令实现对参考文献信息和格式的完全控制。

基于 bibtex 程序的方法总共有四步编译, 第一步 T<sub>E</sub>X 编译程序将信息写入 aux 文件, 第二步 bibtex 程序读取 aux 文件和 bib 文件并将信息写入 bbl 文件, 第三步 T<sub>E</sub>X 编译程序读取 bbl 文件将信息写入 aux 文件, 第四步 T<sub>E</sub>X 编译程序再次读取 aux 文件, 来生成文献表和文献引用标注标签。这在 BibLaTeX 设计者看来是没有必要的, 既然要实现对参考文献信息的完全控制, 那么将参考文献信息放在内存中是最好的方式, 也没有必要过多的读写文件。如果参考文献信息能够以一种合适的方式读入内存, 并能在 tex 宏中应用, 那么就可以解决

问题。

因此 BibLaTeX 设计并实现了这种新的方式, 在第一步编译时, T<sub>E</sub>X 编译程序将需要的信息写入到 bcf 文件中, 然后在第二步编译中, 利用外部程序 Biber 读取该文件, 并处理 bib 文件得到一个 bbl 文件 (其内包含 BibLaTeX 宏包设计的能被 tex 宏读取并利用的固定格式的信息, 这一信息包含了所有的参考文献信息, 可以很方便地用于参考文献著录表和参考文献引用标注的输出)。第三步编译时, T<sub>E</sub>X 编译程序读取 bbl 文件, BibLaTeX 宏包识别并利用其中的信息输出引用标签和文献表。这种新的方式仅用两次次源文档编译和一次外部程序 Biber 编译, 就实现了参考文献生成, 相比传统的基于 BibTeX 程序的方法, 既减少了一次源文档编译, 同时也实现了将全部参考文献信息保存在内存中并进一步利用的目的。

如前所述, 参考文献生成包含两个部分, 一个是著录文献表, 一个是正文中的引用标注。对于 L<sup>A</sup>T<sub>E</sub>X 传统方法, 前者以手工写或者 bibtex 程序结合 bst 文件生成, 后者则由标注宏包如 natbib 等读取手动写入或者 bibtex 程序结合 bst 文件给出的标签信息生成。而 BibLaTeX 由于已经把全部参考文献信息读取到内存中, 因此可以由 tex 宏完全控制。BibLaTeX 提供了所有这些参考文献信息的控制接口, 著录表格式由一系列宏进行控制, 标注格式也由一系列宏进行控制, 前者通常放在 bbx 文件中, 后者则放在 cbx 文件中, 一些通用的宏则放在 biblatex.sty, biblatex.def, standard.bbx, english.lbx, blx-dm.def 等文件中 (在格式修改时可在这些文件中查看原始的格式定义)。

简单来说, 对于著录文献表, BibLaTeX 通过 driver 控制条目中的域 (项) 和顺序, 通过 fieldformat 控制各个项的格式, 其中利用一套异步标点控制机制和一套本地化字符串系统辅助。对于标注标签, 则通过各类 cite 命令及其内部接口进行控制, 同样使用标点控制和本地化字符串机制。此外, BibLaTeX 充分利用 tex 的局部化机制, 提供各种局部化设置方式, 比如利用文境 refcontext 区分不同的数据和排序机制, 利用各 cite 命令和 bibliography 和 biblist 环境区分各种分隔符的输出格式, 利用各类条目类型的驱动 (即 driver) 环境设置域的不同输出格式。可以说, biblatex 因为解决了参考文献全部信息的控制问题, 为利用 tex 宏控制参考文献和引用标签提供了无限可能, 只有想不到, 没有做不到。

### 1.3 biblatex 参考文献生成的优缺点

如前所述, 在源文档编译中 BibLaTeX 宏包通过写 bcf 文件 (格式实际为 xml) 为 biber 提供处理信息, Biber 根据这些信息处理 bib 文件中相关条目, 完成筛选、排序、转换等操作, 并将处理结果写入 bbl 文件中, 再次进行源文档编译时, BibLaTeX 宏包则读取 bbl 中的完整数据以进一步处理输出。BibLaTeX 宏包基于这样一种新的机制实现了更简化的文献编译, 实现了参考文献信息完全由 tex 宏控制, 还实现了其它大量功能来替代传统参考文献生成方法所需的其它辅助宏包。这些功能的实现, 有赖于 BibLaTeX 宏包的全面设计以及后端程序 Biber 的强大支持。总结起来 BibLaTeX 加上其后端程序 Biber 具有如下主要特点:

- BibLaTeX 包含许多用于人文社科类的样式, 用于生成人文社科类的文献表和标注标签。而基于 bibtex 程序的方法引用标注主要包括作者年制和数字顺序编码制, 缺乏人文社科类文献常用的作者标题制或者脚注样式, 需要利用类似 jurabib 这样的宏包以及专门的 bst 文件。
- BibLaTeX 提供便捷的参考文献格式控制机制, 包括友好的宏包选项控制和简单的 tex 宏格式修改, 非常易学。而传统的方法则要么手工改写文献表内容, 要么修改 bst 样式文件, 而学习 bst 文件采用的与 tex 语法迥异的语法并不是一件轻松的事情。
- BibLaTeX 提供完全的 unicode 支持, 提供自动的编码转换, 提供自动的姓名和姓名列表

非模糊处理, 支持可定制的数据继承规则, 提供非常灵活的数据映射 (动态处理)。

- BibLaTeX 提供强大排序功能, 支持排序定制, 支持 unicode 字符的排序, 更支持本地语言的排序调整, 比如中文的按笔画数排序等, 可以实现丰富的效果。而基于 bibtex 的方法能支持 utf-8 编码的 bib 文件, 但不支持非 ASCII 字符的排序。
- BibLaTeX 使用的后端 biber, 具有处理大文献量的能力, 不用担心 bibtex 可能存在的内存问题。biber 还支持远程的参考文献数据库。
- BibLaTeX 兼容 BibTeX 格式外, 还支持更多更广泛的条目类型和域。也支持除 BibTeX 外的更多格式的文献数据库, 也支持其他格式的文献数据的输出和转换。
- BibLaTeX 提供参考文献分节、分类筛选等功能。在文档中打印多个文献表非常方便, 而不再需要使用其它宏包, 比如 chapterbib 等。
- BibLaTeX 是包含 biber 程序的一个大型宏包。BibLaTeX 提供了覆盖 natbib、jurabib、chapterbib、multibib、mcite、inlincite 等宏包的完整功能体系。

当然也不存在十全十美的事物, BibLaTeX 也存在一些局限, 包括:

- 虽然 BibLaTeX 已经取得了广泛应用, 但一些期刊和出版商长期维护使用专门设计的配合 bibtex 程序使用的 bst 样式文件, 因此可能不接收使用 BibLaTeX 的文档。
- BibLaTeX 生成处理完的参考文献信息, 不是 thebibliography 环境, 而是 BibLaTeX 预设的格式, 因此其代码并不能像 thebibliography 环境中的内容一般可以直接拷贝进源文档使用。
- BibLaTeX 还在不断更新和升级, 尽管已经基本稳定, 但可能存在后向兼容问题。比如随着 BibLaTeX 的升级, 一些基于老版本 BibLaTeX 的第三方样式可能会无法使用, 而基于新版本 BibLaTeX 的样式可能无法应用于老版本 BibLaTeX 的环境, 这需样式作者持续维护。
- BibLaTeX 因为功能强大且全面, 说明文档很大很全, 有点类似技术说明书, 但作为初级用户的入门却并不合适, 而对于宏包作者这一类的用户, BibLaTeX 宏包的代码实现和解析说明相对较少, 也不利于这类人员深入学习和参与维护。

从前述的优缺点可以看到, 虽然 BibLaTeX 宏包还并不完美, 但基于 BibLaTeX 生成参考文献的确可以获得很多优势, 因此对于一般用户来说, 使用 BibLaTeX 不失为一种好的选择。

## 2 参考文献常见需求与解决方案

### 2.1 参考文献生成的常见需求

L<sup>A</sup>T<sub>E</sub>X 中文写作对于参考文献一般会有哪些需求? 最直接地, 我们会想到文章投稿时的参考文献格式要求, 比如: 参考文献标题的字体字号前后间距段落格式; 参考文献内容的字体字号; 各条目的间距; 条目的内容格式如文献标识符、标点等; 双语参考文献; 上标、带年份、带页码等标注要求等等。在书籍之类的文档排版中有时也会遇到比如: 分章参考文献表; 文后参考文献总表; 参考文献的目录链接; 脚注中引用参考文献; 脚注中直接显示参考文献; 在合适位置按需插入文献表等等。还有一些比如文献的排序; PDF 文档中标注和文献条目的正反超链接等。总结起来可能包括如下常见需求:

- 参考文献表基本生成方法-数据源文件准备 (手动、利用 winedt、利用 texstudio、Jabref、随 L<sup>A</sup>T<sub>E</sub>X 文档生成)



- 参考文献表基本生成方法-文档的组成 (基本结构、宏包加载、样式包加载、文献引用、文献表打印)
- 参考文献表基本生成方法-文档的编译 (手动、利用 `winedt`、利用 `texstudio`、脚本)
- 书后参考文献和分章参考文献
- 分类筛选、缩略信息和索引打印
- 参考文献标题格式 (重定义 `heading`, 利用 `titlesec`)
- 参考文献内容格式 (重定义内容命令)
- 参考文献著录样式 (顺序编码制、作者年制、域、位置、标点、排序)
- 参考文献标注样式 (页码、年份、对应 `natbib/mcitem` 等的标注命令)
- 多语言文献
- 脚注题注小页环境中的引用
- 脚注旁注中的文献表
- `beamer` 类中的参考文献

下面针对前述提出的需求,一一进行解答。解决方案主要应用于一般文档类 (`book`, `report`, `article`), `beamer` 类的问题放到最后一小节中专门介绍。

## 2.2 参考文献数据文件准备

L<sup>A</sup>T<sub>E</sub>X 文档中生成参考文献除了利用 `thebibliography` 环境手工输入参考文献条目信息这一简单方式外, `bibliography+bibliographystyle+bibtex` 方式或者 `biblatex` 方式都需要准备一个参考文献数据源文件即 `bib` 文件。(这里需要区分一下的是 `bibtex` 程序和 BibTeX 格式, `bibtex` 程序是前面介绍过的传统参考文献生成方法中用于生成 `thebibliography` 环境信息的外部程序, 类似于 BibLaTeX 宏包提供的 `biber` 程序, 这个程序并不是唯一的, 大家都可以写一个这样的程序来实现类似的功能, 比如笔者用 `python` 写的 `bibmap` 程序, 也能够输出 `thebibliography` 环境来生成参考文献, 只是参考文献格式定义不使用 `bst` 而使用 `py` 文件罢了。而 BibTeX 格式则是参考文献数据源 `bib` 文件的内部组织格式, 是通用的, BibLaTeX 宏包和 `biber` 程序同样使用这种格式的文件, `bibtex` 程序、`biber` 程序同样也使用。)

`bib` 文件内保存有各条参考文献的信息, 每一条参考文献记录都是一个 BibTeX item (BibTeX 项), 一条参考文献记录基本格式如下:

示例 2 参考文献信息基本格式即 BibTeX 格式

```
1 @ARTICLE{entrykey,
2   author = {Author A and Author B and Author C and others},
3   title = {A title of a paper},
4   journal = {A title of a journal},
5   year = {2005},
6   volume = {31},
7   number = {6},
8   pages = {75-81},
9 }
```

代码

其中`@`符号后面紧跟着条目的类型比如 `article`(期刊文章), `book`(专著) 等, 编组符号 `{` 后紧跟的是该条参考文献信息的 `bibtex key` 即引用关键词, 在 `cite` 标注 (引用) 命令中使用该关键词来表示对此文献的引用。逗号后面跟着的是构成参考文献记录的各个域的信息, 比如作者域, 题名域等等, 参考文献信息放在各个域名等号后面的编组内, 一些域 (比如 `author` 域) 对于信息录入是有特殊格式要求的 (比如英文姓名的两种不同方式), 具体可以参考 `biblatex` 宏包手册和 `biblatex-gb7714-2015` 样式包手册<sup>[10]</sup> 中关于域信息录入的说明。

尽管 biblatex 在传统 BibTeX 格式基础上增加了很多新的条目类型和域, 但它完全兼容传统的 BibTeX 格式文件, 出于兼容性考虑, 我们在准备 bib 文件时, 可以完全按照**传统的 BibTeX 格式规范**来进行。BibTeX 格式中有两个基本概念: 一是条目类型 (Entry type), 即一个参考文献条目的类型, 是 book(专著), 是 inbook(专著析出的文献), 或其它类型。二是域 (Field), 即纪录参考文献信息并组成参考文献条目的信息项, 比如 author(作者), title(题名), 或者其它。BibTeX 格式中条目类型和域如表1、2所示, 这些条目和域是 bibtex 附带的默认样式 bst 文件能识别的条目类型和域, 而第三方的 bst 文件有可能支持其它条目类型和域, 比如一些 bst 样式为了兼容 biblatex 增加的条目类型和域而做的处理。biblatex 虽然支持更多的条目类型和域 (详见 biblatex 手册), 但通常情况下 BibTeX 默认的条目类型和域已够用。

表 1 BibTeX 格式的条目类型和域

条目类型	类型说明	条目必选域	条目可选域
article	An article from a journal or magazine. author, title, journal, year, volume	number, pages, month, note, key	
book	A book with an explicit publisher.	author/editor, publisher, year	title, volume/number, series, address, edition, month, note, key, url
booklet	A work that is printed and bound, but without a named publisher or sponsoring institution.	title	author, howpublished, address, month, year, note, key
inproceedings / conference	An article in a conference proceedings.	author, title, booktitle, year	volume/number, series, pages, address, month, organization, publisher, note, key
inbook	A part of a book, usually untitled. May be a chapter (or section, etc.) and/or a range of pages.	author/editor, chapter/pages, publisher, year	title, volume/number, series, type, address, edition, month, note, key
incollection	A part of a book having its own title.	author, title, booktitle, publisher, year	editor, volume/number, series, type, chapter, pages, address, edition, month, note, key
proceedings	The proceedings of a conference.	title, year	editor, volume/number, series, address, month, publisher, organization, note, key
manual	Technical documentation.	title	author, organization, address, edition, month, year, note, key
mastersthesis	A Master's thesis.	author, title, school, year	type, address, month, note, key
phdthesis	A Ph.D. thesis.	author, title, school, year	type, address, month, note, key
techreport	A report published by a school or other institution, usually numbered within a series.	author, title, institution, year	type, number, address, month, note, key
misc	For use when nothing else fits.	none	author, title, howpublished, month, year, note, key
unpublished	A document having an author and title, but not formally published.	author, title, note	Optional fields: month, year, key

参考文献源文件本质上只是一个文本文件, 只是其内容需要遵守 BibTeX 格式, 所以可以有多种生成方式:

表 2 BibTeX 格式中的域说明

域名称	域说明
address	Publisher's address (usually just the city, but can be the full address for lesser-known publishers)
annote	An annotation for annotated bibliography styles (not typical)
author	The name(s) of the author(s) (in the case of more than one author, separated by and)
booktitle	The title of the book, if only part of it is being cited
chapter	The chapter number
crossref	The key of the cross-referenced entry
edition	The edition of a book, long form (such as "First" or "Second")
editor	The name(s) of the editor(s)
howpublished	How it was published, if the publishing method is nonstandard
institution	The institution that was involved in the publishing, but not necessarily the publisher
journal	The journal or magazine the work was published in
key	A hidden field used for specifying or overriding the alphabetical order of entries (when the "author" and "editor" fields are missing). Note that this is very different from the key (mentioned just after this list) that is used to cite or cross-reference the entry.
month	The month of publication (or, if unpublished, the month of creation)
note	Miscellaneous extra information
number	The "(issue) number" of a journal, magazine, or tech-report, if applicable. (Most publications have a "volume", but no "number" field.)
organization	The conference sponsor
pages	Page numbers, separated either by commas or double-hyphens.
publisher	The publisher's name
school	The school where the thesis was written
series	The series of books the book was published in (e.g. "The Hardy Boys" or "Lecture Notes in Computer Science")
title	The title of the work
type	The field overriding the default type of publication (e.g. "Research Note" for techreport, "PhD dissertation" for phdthesis, "Section" for inbook/incollection)
volume	The volume of a journal or multi-volume book
year	The year of publication (or, if unpublished, the year of creation)

### 2.2.1 手动文本文件生成

利用文本编辑器比如 notepad++ 等, 生成或者编辑 bib 源文件是一种很简单的方式, 直接新建 bib 文件, 用 notepad++ 打开, 填入需要的参考文献信息, 保存就可以得到一个 bib 源文件, 如图 1 所示。notepad++, notepad2 等文本编辑器均可在其官网下载到。

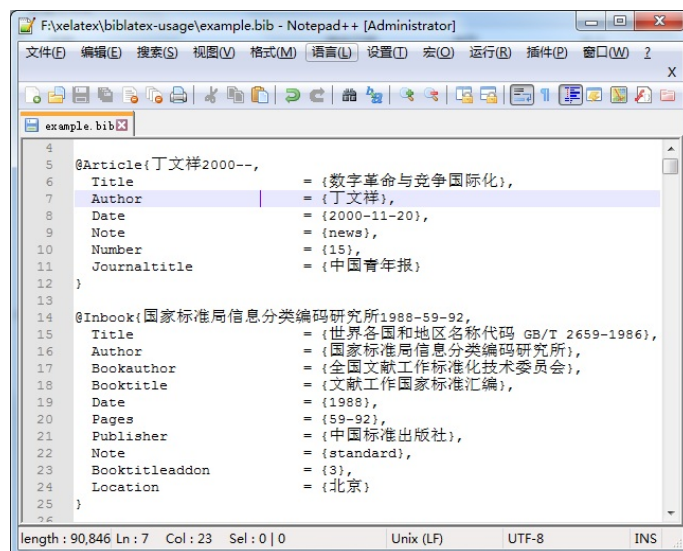


图 1 文本编辑器生成 bib 文件



### 2.2.2 利用 winedt 生成

在 windows 下常用 winedt 编辑 latex 文档, 所以用其生成 bib 文件也是一种不错的方式, 特别是它提供了各种条目类型的信息模板 (图2)。winedt 可以在其官网 (<http://www.winedt.com/>) 下载到, 非注册版本不影响功能使用。

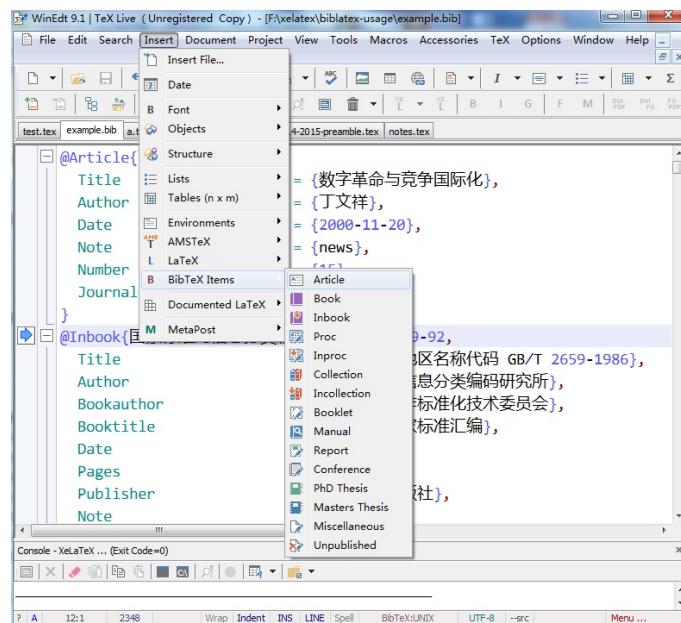


图 2 winedt 生成 bib 文件

### 2.2.3 利用 texstudio 生成

texstudio 软件可以跨平台使用, linux 下因为没有 winedt 所以比较常用, 用其生成 bib 文件也很方便, 它也提供各种条目类型的信息模板 (图3)。texstudio 软件下载地址: <https://sourceforge.net/projects/texstudio/?source=navbar>。

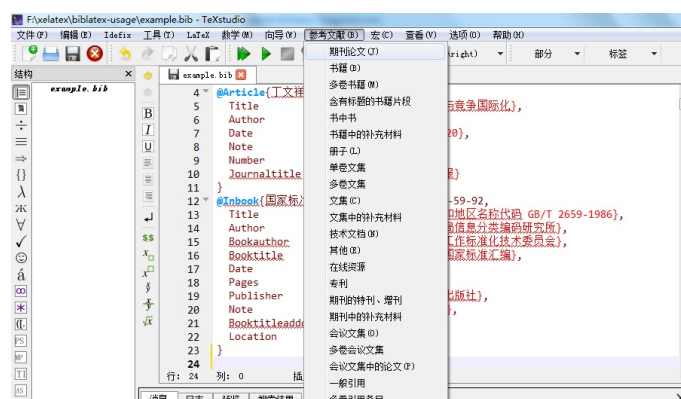


图 3 texstudio 生成 bib 文件

### 2.2.4 利用 Jabref 软件生成

利用 Jabref 软件是最重要的一种方式, 因为它也是一个较强的参考文献管理工具。新建一个数据库就是新建一个 bib 文件 (图4), 新建一个记录就是加入一条参考文献信息 (图5), 参考文献的信息的录入也是可视化输入栏形式 (图6)。jabref 软件可在官网 (<http://www.jabref.org/>) 下载。

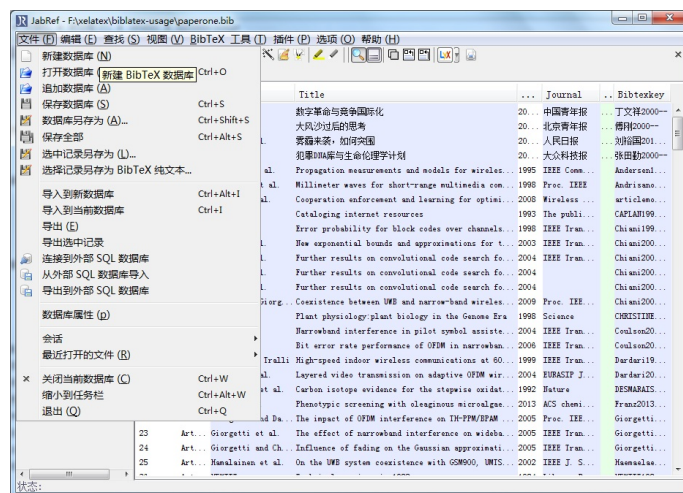


图 4 Jabref 软件生成 bib 文件

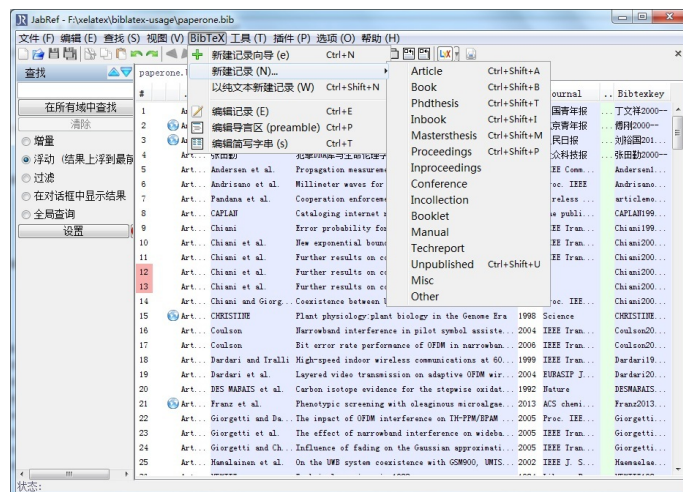


图 5 Jabref 软件增加参考文献条目

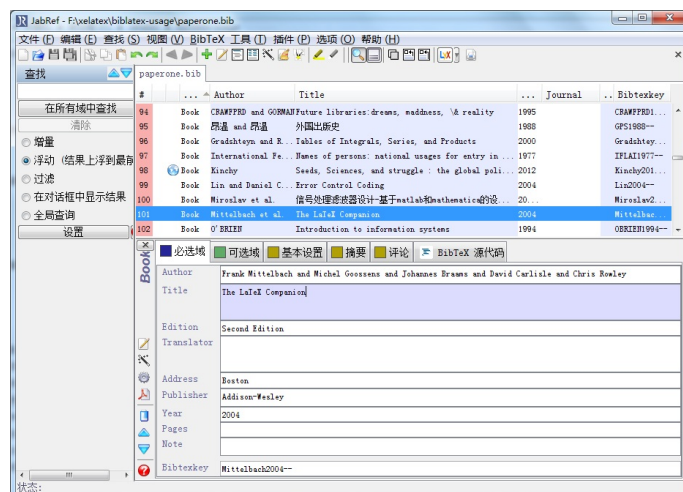


图 6 Jabref 软件录入参考文献信息

类似工具还有：Papers<sup>2</sup>、Mendeley<sup>3</sup>、zotero<sup>4</sup>等。

<sup>2</sup><http://papersapp.com>

<sup>3</sup><http://www.mendeley.com>

<sup>4</sup><https://www.zotero.org/>

### 2.2.5 随 L<sup>A</sup>T<sub>E</sub>X 文档生成

L<sup>A</sup>T<sub>E</sub>X 提供了 filecontents 环境允许在 L<sup>A</sup>T<sub>E</sub>X 文档编译的时候自动将信息写入文件中, 自然这也可以用来生成 bib 文件。比如在导言区加入如下代码, 可以生成一个 example.bib 文件。

示例 3 利用 filecontents 环境随 latex 文档生成 bib

代码

```
1 \usepackage{filecontents}
2 \begin{filecontents}{example.bib}
3 @Article{傅刚2000--,
4   Title = {大风沙过后的思考},
5   Author = {傅刚 and 赵承 and 李佳路},
6   Note = {news},
7   Number = {14},
8   Url = {http://www.bjyouth.com.cn/Bqb/20000412/GB/4216%5ED0412B1401.htm},
9   Urldate = {2005-07-12},
10  Date = {2000-04-12},
11  Journaltitle = {北京青年报}
12 }
13 \end{filecontents}
```

### 2.2.6 获取标准 bib 文件

上述给出的方法都是手动编写 bib 文件的方法, 事实上这些手工编写方法还是比较繁琐的。那么是否有其它方法可以直接获取 bib 文件或其内容避免自己手工编写呢? 答案是有的, latexstudio 上共分享了这方面的几篇文章<sup>[12][13][14]</sup>。总结起来主要有几个渠道:

- (1) Google 学术<sup>5</sup>: 检索文献 » Cite (引用) » BibTeX » 复制粘贴到 mybibtex.bib 文件
- (2) CNKI, (以 firefox 为例): (确保安装 Zotero 的 firefox 扩展 ») CNKI 检索文献 » 点击一下 url 输入框末尾的文件夹图标 » 选中要导出的文献 » 导出为 BibTeX 引用格式 » 复制粘贴导出的 BibTeX 文献到 mybibtex.bib 文件。
- (3) 必应学术 (必应学术-学无止境, 术有乾坤)、百度学术 (百度学术-保持学习的态度) 类似于 google 学术。
- (4) CiteSeerX 搜索引擎<sup>6</sup>。
- (5) 从一些学术期刊官网直接获取 bib 文件, 比如 IEEE, Springer, Elsevier 等
- (6) 使用文献助手工具, 地址: <http://cite.latexstudio.net>

熟练使用这些方法, 应该可以进一步提高工作效率。说明一下: latexstudio 网站原创、收集、整理了很多 latex 方面的知识、信息和资源, 内容不一, 对不同程度的 latex 爱好者都具有一定的参考价值, 各位朋友可以了解学习。

### 2.2.7 从 pdf 或文本文件转换 bib 文件

批量生成参考文献数据以构成 bib 文件, 还有一种方式是从参考文献表文本转换。比如从一个包含 GB/T 7714-2015 格式的参考文献表的 pdf 文件中, 将文献表拷贝到一个文本文件中 (比如: textfilename.txt 文件), 其中每一条文献信息构成一个行。然后利用 biblatex-gb7714-2015 样式宏包附带的 perl 脚本 gb7714texttobib.pl 将其转换为 bib 文件 (比如: bibfilename.bib)。则对应的脚本使用命令为:

<sup>5</sup><https://scholar.google.com>

<sup>6</sup><http://citeseerx.ist.psu.edu>

#### 示例 4 利用 gb7714texttobib.pl 将参考文献文本转换为 bib 文件

代码

```
1 perl gb7714texttobib.pl in=textfilename.txt out=bibfilename.bib
```

其中 in= 后面是输入的参考文献文本文件，out= 后面是输出的 bib 文件。

#### 2.2.8 利用 biber 转换输出新的 bib 文件

对于使用 biblatex 的用户来说，当需要从一个大的 bib 数据库中抽取部分文献数据到一个新的 bib 文件时，可以利用 biber 命令选项 `--output-format=bibtex`。事实上，因为平时的积累我们通常会维护一个大的参考文献数据库，但写文章的时候只是引用其中的部分文献，当需要提交文章时，我们提交仅包含引用文献数据的 bib 文件即可，这可以通过 biber 轻松完成。例 5 给出了一个 biber 命令示范，其中 jobname 是当前引用文献的 tex 文档名，运行该命令后会得到一个 jobname\_biber.bib 的文件，包含有该文档所引用的文献的数据。

#### 示例 5 利用 biber 转换出新的 bib 文件

代码

```
1 biber --output-format=bibtex jobname.bcf
```

#### 2.2.9 利用 bibmap 程序修改 bib 文件

对于 bib 文件经常会有批量处理需求，一是从一个大的 bib 数据库，提取少量的要引用的文献；二是对 bib 文件中的条目内容进行修改。对于这两个需求，笔者利用 python 重写了 bib 文件解析器，并借鉴 biblatex 的动态数据修改的模式，实现了一个 bib 文件快速修改的工具。因为数据修改的基本逻辑与 biblatex 的类似，所以命名为 bibmap。具体使用方法见其说明文档。

### 2.3 L<sup>A</sup>T<sub>E</sub>X 文档的组成

L<sup>A</sup>T<sub>E</sub>X 文档的源文件是由具有某种结构和模式的代码所构成的，需要生成参考文献表的 L<sup>A</sup>T<sub>E</sub>X 文档自然也遵守这样的结构和模式。

#### 2.3.1 文档源文件基本结构

L<sup>A</sup>T<sub>E</sub>X 文档源文件的代码基本结构由导言区和正文区构成，导言区位于 documentclass 命令与 begin{document} 命令之间，而正文区位于 begin{document} 命令和 end{document} 命令之间。宏包在导言区加载，具体文档内容在正文区编辑。

#### 示例 6 文档源文件的代码基本结构

代码

```
1 \documentclass{article}%文档类
2
3 %导言区:
4 %ctex中文设置
5 \usepackage{ctex}
6 %定义版面
7 \usepackage[paperwidth=210mm,paperheight=290mm,left=20mm,right=20mm,top=25mm,bottom=15mm]{geometry}
8 %书签功能, 选项去掉链接红色方框
9 \usepackage[colorlinks=true,pdfstartview=FitH,linkcolor=blue,anchorcolor=violet,citecolor=magenta]{
    hyperref}
10
11 \begin{document}%正文区开始
12 正文内容
13 \end{document}
```

### 2.3.2 biblatex 宏包和样式的加载、指定参考文献数据源

利用 biblatex 宏包生成参考文献可以说非常简单, 只要在上述 L<sup>A</sup>T<sub>E</sub>X 基本文档基础上载入 biblatex 宏包, 选择需要的参考文献样式, 并指定 bib 数据源, 然后在正文中引用 bib 文件中的参考文献, 然后在文档合适位置 (比如文档末尾) 打印参考文献表。例7和图7给出了示例代码及其结果。

示例 7 biblatex 参考文献生成最小工作示例

代码

```

1 \documentclass{article}%文档类
2 %导言区开始:
3 %加载ctex宏包, 中文支持
4 \usepackage{ctex}
5 %加载geometry宏包, 定义版面
6 \usepackage[left=20mm,right=20mm,top=25mm,bottom=15mm]{geometry}
7 %加载hyperref宏包, 使用超链接
8 \usepackage[colorlinks=true,pdfstartview=FitH,linkcolor=blue,anchorcolor=violet,citecolor=magenta]{
  hyperref}
9 %加载biblatex宏包, 使用参考文献
10 %标注(引用)样式citestyle, 著录样式bibstyle都采用gb7714-2015样式
11 \usepackage[bibstyle=gb7714-2015,%nature,%
12 citestyle=gb7714-2015%,backref=true%
13 ]{biblatex}
14 %biblatex宏包的参考文献数据源加载方式
15 \addbibresource[location=local]{example.bib}
16
17 %正文区开始:
18 \begin{document}
19 %正文内容, 引用参考文献
20 详见文献\cite{Peebles2001-100-100}\parencite{Miroslov2004--}
21 参考文献\cite{见}[49页]{蔡敏2006--}\parencite{见}[49页]{Miroslov2004--}
22
23 %打印参考文献表
24 \printbibliography[heading=bibliography,title=参考文献]
25 \end{document}

```

详见文献<sup>[1]</sup>[2] 参考文献<sup>[见 3, 49 页]</sup>[见 2, 49 页]

#### 参考文献

- [1] Von PEEBLES P Z Jr. Probability, random variable, and random signal Principles and L<sup>A</sup>T<sub>E</sub>X[M]. 4th ed. New York: McGraw-Hill, 2001: 100 (引用页: 13).
- [2] MIROSLAV D L, DEJAN V T, BRIAN L E. 信号处理滤波器设计-基于 matlab 和 mathematica 的设计方法[M]. 朱义胜, 董辉, 译. 北京: 电子工业出版社, 2004 (引用页: 13).
- [3] 蔡敏, 徐慧慧, 黄柄强. UML 基础和 Rose 建模教程[M]. 北京: 人民邮电出版社, 2006 (引用页: 13).

图 7 最小工作示例的结果

其中, biblatex 宏包和参考文献数据源都在导言区加载 (例8), 宏包加载命令为 usepackage, 数据源指定命令为 addbibresource。

示例 8 biblatex 宏包、数据源和样式的加载

代码

```

1 %biblatex宏包加载
2 %其中后端backend使用biber
3 %引用样式citestyle, 著录样式bibstyle都采用gb7714-2015样式

```



---

```

4 \usepackage[backend=biber,bibstyle=gb7714-2015,%nature,%
5 citestyle=gb7714-2015%,backref=true%
6 ]{biblatex}
7
8 %参考文献数据源加载
9 \addbibresource[location=local]{example.bib}

```

---

其中, 参考文献样式作为 biblatex 宏包的可选参数加载, 即用 bibstyle 选项加载著录样式, 用 citestyle 选项加载标注样式, 当两者使用同名样式时, 可以用一个选项 style 表示, 例8中 bibstyle 和 citestyle 都选择使用 gb7714-2015 样式。有时标注样式和著录样式单独指定会很有用的, 比如有人可能想用 gb7714-2015 的参考文献著录样式, 但引用标注又不想要用上标? 最简单的方式是著录样式使用 gb7714-2015 样式, 而标注样式使用其他样式比如 author-year。当然修改 gb7714-2015 的标注样式文件中 cite 等命令也可以实现相同目的。

可用的参考文献样式还有许多, 既可以使用 biblatex 宏包提供的标准样式 (详见 biblatex 手册 Standard Styles 一节), 也可以使用第三方提供的定制样式, 比如 nature, science 等, 具体参见节, 用户可以根据需要选择使用。

为利用 biblatex 的高级功能, 后端 backend 应尽量采用 biber 而不用 bibtex (如果 backend 为 bibtex8, 则 biblatex 宏包利用 bibtex8 程序和 aux 文件进行编译, 而 backend 为 biber 则是利用 biber 程序和 bcf 文件进行编译)。

另外, biblatex 宏包还提供一系列宏包选项, 用于控制文献著录表, 标注的标签等内容, 详细内容参考 biblatex 手册宏包选项一节。

值得注意的是: 参考文献数据库源文件即 bib 文件可以是本地的, 也可以是网络上的 (具体参考 biblatex 手册 Bibliography Commands 一节中关于 addbibresource 命令的说明)。本地的数据源可以指定绝对路径或者相对路径, 注意其中目录层级之间的间隔符用/而不是\, windows 下默认复制的间隔符是\, 需要将其改为/。数据源还可以指定标签, 以便在不同参考文献分节中使用不同的文献源。数据源添加方式, 见例9。

#### 示例 9 添加参考文献数据源的两种方式

代码

---

```

1 \addbibresource{bibfile1.bib} %本地数据源
2 \addbibresource{bibfile2.bib}
3 \addbibresource[location=local]{D:/zlatexreference/paperone.bib}
4 \addbibresource[location=remote]{http://www.citeulike.org/bibtex/group/9517} %网络数据源
5 \addbibresource[location=remote,label=lan]{ftp://192.168.1.57/~user/file.bib}
6
7 %指定文献源标签
8 \addbibresource[label=drofnats]{./bibfiles/drofnats.bib}
9 \addbibresource[label=frobwangler]{./bibfiles/frobwangler.bib}
10 \addbibresource[label=everyone]{./bibfiles/everyone.bib}
11
12 %在不同的文献节中使用不同的标签的文献源
13 \newrefsection[drofnats,everyone]
14 section contents
15 \endrefsection
16
17 \begin{refsection}[frobwangler,everyone]
18 section contents
19 \end{refsection}

```

---

### 2.3.3 文献的引用（标注）

文献的引用（标注）可以采用 L<sup>A</sup>T<sub>E</sub>X 提供的 cite 命令（注意 biblatex 对其进行了重定义），也可以使用 biblatex 提供的其它命令比如 parencite、textcite 等，也可以采用参考文献样式包提供的定制命令，比如 gb7714-2015 样式提供的 pagescite、yearpagescite 等。在文档中引用文献仅需要在合适的位置插入引用（标注）命令，命令的必选参数是所引用参考文献的 bibtex 键（即引用关键词）。比如：

示例 10 参考文献的引用命令

代码

```
1 详见文献\cite{Peebles2001-100-100}\parencite{Miroslav2004--}
2 参考文献\cite[见][49页]{蔡敏2006--}\parencite[见][49页]{Miroslav2004--}
3 详见文献\pagescite{Peebles2001-100-100}\pagescite[][201-301]{Peebles2001-100-100}
4 见赵耀东\yearpagescite[][205]{赵耀东1998--}和Simon\yearpagescite[][15]{Simon2001--}的文献。
```

当需要在参考文献表中打印不标注的文献时，可以使用 nocite 命令引入这些文献。其参数是具体条目的引用关键词（bibtex 键）或者星号，星号表示将 bib 文件中所有的参考文献引入。使用命令如 11 所示。

示例 11 不标注的文献的引用命令

代码

```
1 参考文献表中打印不标注的文献引入\nocite{Miroslav2004--，蔡敏2006--}。
2 参考文献表中打印不标注的文献引入，引入bib文件中全部文献\nocite{*}。
```

另外，在 biblatex 中使用动态文献集也能产生类似 nocite 命令的效果，即打印文献表时也会自动输出定义在动态文献集中的文献。

### 2.3.4 文献表的打印（输出）

基于 biblatex 宏包的参考文献表打印（输出）与 L<sup>A</sup>T<sub>E</sub>X 传统方式不同，采用的命令是 printbibliography。如果全文仅需一个参考文献，那么只要在合适的地方插入如下命令即可：

示例 12 参考文献表打印命令

代码

```
1 \printbibliography[heading=bibliography,title=参考文献]
```

注意：printbibliography 有很多可选参数，用于复杂的文献表控制，具体见 biblatex 宏包手册。

### 2.3.5 参考文献正反超链接

当加载 hyperref 宏包后，biblatex 宏包会自动提供正向超链接，若需要实现的反向超链接，则需要添加 backref 选项，如例 13 所示。

示例 13 参考文献反向超链接选项

代码

```
1 \usepackage[backend=biber,bibstyle=gb7714-2015,citestyle=gb7714-2015,%
2 backref=true%
3 ]{biblatex}
```

需要注意的是，反向超链接主要是对页码进行链接，而对页码进行正确超链接需要一个前提，即当前链接的页码是唯一的。当有两页具有相同的页码时，超链接总是链接到第一个具有该页码的页面上。比如当封面页页码是 1，正文第一页页码也是 1，那么到正文第一页的反向超链接总会链接到封面页上。解决该问题就是要解决页码的唯一性问题。当页码的表示形式不同时，页码也能得到正确的区分。比如封面页用罗马字母 I 表示，而正文第一页用阿拉伯数字 1 表示，那么就能得到正确的反向超链接。

测试文档见：[test-backref.tex](#)。

## 2.4 文档的编译

文档的编译与 L<sup>A</sup>T<sub>E</sub>X 传统方法一致, 只是中间的参考文献编译过程略有变化, biblatex 宏包使用 biber 后端时, 编译参考文献数据用的命令为 biber filename。编译命令可以利用软件调用, 也可以在命令行手工输入。后面将介绍利用 winedt、texstudio、命令行和脚本等不同方式进行编译的具体操作过程。

需要注意, 无论采用哪种方式编译, 如果参考文献数据中存在一些特殊符号比如 & 等, 编译时会报错。对于这个问题除了手动修改 bib 文件中参考文献信息这一种方法外, 还有两种自动处理方式, 一是利用 jabref, jabref 有 bibtex 和 biblatex 两种模式, 在 biblatex 模式下, 对于一些特殊字符的输入, 比如 &, jabref 会自动将其转换为 \&。二是利用 biber 等工具自动处理, 比如使用 gb7714-2015 样式, 在 bib 数据源不经任何处理的情况下, 由 gb7714-2015 样式的设定, biber 工具也能自动处理一些特殊字符。

### 2.4.1 编译命令

一般情况下采用 xelatex 或 lualatex 进行编译, 命令为:

示例 14 xelatex 编译命令

```
1 xelatex jobname.tex
2 biber jobname
3 xelatex jobname.tex
4 xelatex jobname.tex
```

代码

当然也可以使用 pdflatex 进行编译, 命令为:

示例 15 pdflatex 编译命令

```
1 pdflatex jobname.tex
2 biber jobname
3 pdflatex jobname.tex
4 pdflatex jobname.tex
```

代码

但要注意的是: 使用 pdflatex 进行编译, 可能因为某些样式比如 authoryear, 使用了 xstring 宏包中的一些命令, 进而导致错误, 但 numeric 类样式通常没有问题。这一问题在 biblatex 更新到 3.12 版本后取消 xstring 宏包后得以解决。

中文用户编译还需要注意编码问题。utf-8 编码的文档, 采用 xelatex 编译没有任何注意事项, 但使用 pdflatex 编译时, 需要给 ctex 宏包加载 UTF8 选项, 比如 \usepackage[UTF8]{ctex}, 该选项在文档类加载时给出也可, 比如 \documentclass[[UTF8]{article}, 同时引用文献时使用的引用关键词必须是英文, 若使用中文将报错。

当文档使用其他编码时, xelatex 编译通常需要指定一个文档编码, 比如 windows 环境下的 GBK/GB2312 编码的文档需要指定 \XeTeXinputencoding "GBK", 否则会显示乱码。使用 pdflatex 进行编译时, 如果 biblatex 不能正确的处理编码问题, 那么需要为其明确的指定 texencoding 和 bibencoding 选项。比如 windows 环境下的 GBK/GB2312 编码的文档, 需要指定 \usepackage[texencoding=GBK]{biblatex}。

参考测试文档: [test-gbk-pdflatex.tex](#), [test-gbk-xelatex.tex](#), [test-utf8-pdflatex-cls.tex](#), [test-utf8-pdflatex-pkg.tex](#)。


另外也可以利用 latexmk 宏包实现一步编译, 即使用一条命令即可完成整个编译过程, 因为 latexmk 会根据文档的内容自动的设置编译步骤并调用相应的命令进行编译, 因此编译命令也可以写为:

示例 16 latexmk 编译命令

```
1 latexmk -xelatex jobname.tex
```



代码

### 2.4.2 利用 winedt 编译

当文档准备好之后, 第一步点击 winedt 工具栏的 xelatex 按钮 ( 图中红框内的按钮) 完成第一遍 L<sup>A</sup>T<sub>E</sub>X 编译; 第二步, 打开命令行 (可以用 winedt 菜单 accessories 下 command prompt 项) 输入命令 biber filename 完成参考文献编译; 第三步, 再次点击 xelatex 编译按钮完成编译, 如果存在反向超链接, 那么需再次用 xelatex 编译一遍才可得到正确结果。

由于可以使用 latexmk 进行一步编译, 因此如果在 winedt 操作界面上增加一个功能按钮, 该按钮对应的命令为 latexmk, 那么只要点击一下按钮就可以实现一键编译<sup>7</sup>, 具体参见: [Winedt 一键编译 LaTeX 文档](#)。

### 2.4.3 利用 texstudio

texstudio 可以跨平台使用, 在 windows 和 linux 下均可。texstudio 可以通过利用元命令构建编译命令实现一键编译, 进入选项菜单的设置 texstudio, 首先在常规选项卡下设置语言为 zh\_cn 将 texstudio 界面语言转变为中文, 然后在构建选项卡设置默认的编译器为 xelatex, 设置默认文献工具为 biber(图8), 然后点击构建并查看的配置按钮 () , 设置工具栏构建并查看按钮 () 对应的命令由 5 个元命令构成 (图9)。如此, 在编译时只要点击: 构建并查看按钮即可完成一键编译功能。其编译的过程提示信息如例17所示 (环境为 windows 7 x64+texlive 2016+tex studio 2.10.8)。

示例 17 texstudio 一键编译过程提示

代码

```

1 开始 : xelatex.exe -synctex=1 -interaction=nonstopmode "test".tex
2 完成
3
4 开始 : biber.exe "test"
5 INFO - This is Biber 2.5
6 INFO - Logfile is 'test.blg'
7 INFO - Reading 'test.bcf'
8 INFO - Found 0 citekeys in bib section 0
9 INFO - Found 4 citekeys in bib section 1
10 INFO - Found 0 citekeys in bib section 0
11 INFO - Processing section 1
12 INFO - Looking for bibtex format file 'example.bib' for section 1
13 INFO - Decoding LaTeX character macros into UTF-8
14 INFO - Found BibTeX data source 'example.bib'
15 WARN - Overwriting field 'year' with year value from field 'date' for entry ' 錄橫搗炳013--'
16 WARN - Overwriting field 'year' with year value from field 'date' for entry ' 鑛紆2011--'
17 WARN - BibTeX subsystem: C:\Users\ADMINI-1\AppData\Local\Temp\pru9Tr4B7_\example.bib_6592.utf8, line 1198,
    warning: possible runaway string started at line 1197
18 INFO - Overriding locale 'en-US' defaults 'normalization = NFD' with 'normalization = prenormalized'
19 INFO - Overriding locale 'en-US' defaults 'variable = shifted' with 'variable = non-ignorable'
20 INFO - Sorting list 'none/global/' of type 'entry' with scheme 'none' and locale 'en-US'
21 INFO - No sort tailoring available for locale 'en-US'
22 INFO - Writing 'test.bbl' with encoding 'UTF-8'
23 INFO - Output to test.bbl
24 INFO - WARNINGS: 3
25 完成
26
27 开始 : xelatex.exe -synctex=1 -interaction=nonstopmode "test".tex
28 完成
29
30 开始 : xelatex.exe -synctex=1 -interaction=nonstopmode "test".tex
31 完成

```

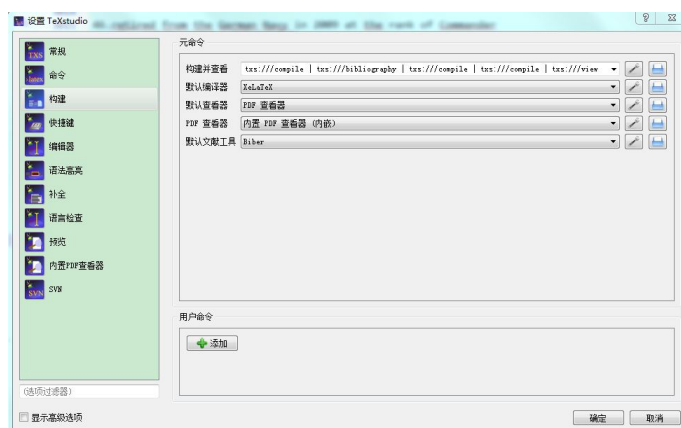


图 8 texstudio 设置默认的编译工具

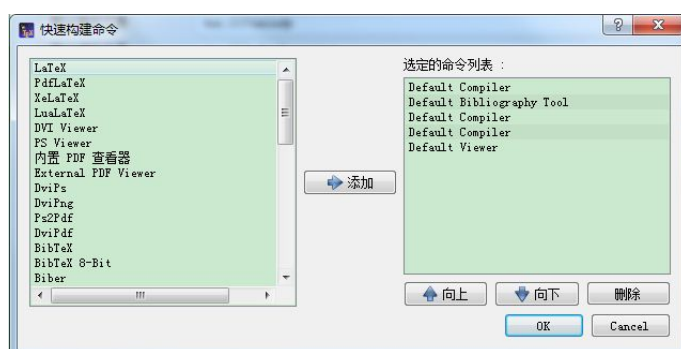


图 9 texstudio 配置构建并查看命令

linux(以 deepin linux x64 v15+texlive 2016+texstudio 为例) 下 texstudio 的设置是类似的, 但要正常编译还有一个关键问题, 即设置编译工具的路径, 这不是系统环境的路径设置, 而是要告诉 texstudio 编译工具所在的目录, 这同样在菜单选项下的 texstudio 设置项中设置, 仍然进入构建选项卡, 左下角点选显示高级选项, 然后设置命令 (\$PATH) 为 /usr/local/texlive/2016/bin/x86\_64-linux, 如图10所示。

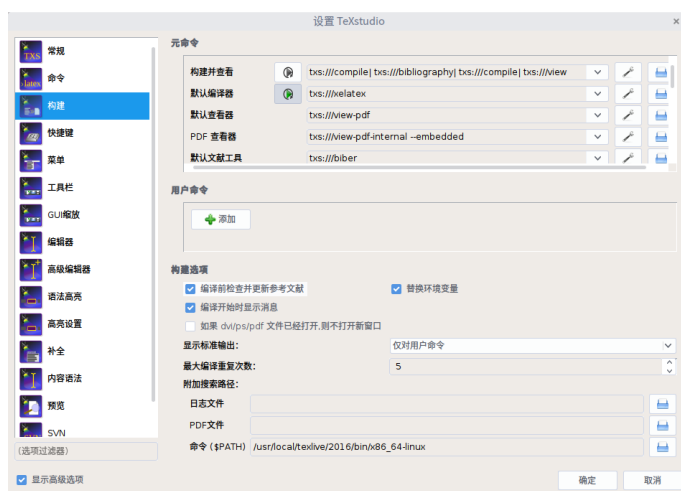


图 10 linux 下 texstudio 设置命令路径

<sup>7</sup><https://blog.csdn.net/xenonhu/article/details/80010659>



### 2.4.4 命令行或脚本

命令行编译本质上是在命令行手动输入命令进行编译。过程仍然是一遍 xelatex 编译, biber 编译, 两遍 xelatex 编译, 命令如下:

示例 18 命令行手动输入的命令

代码

```
1 xelatex.exe -no-pdf -interaction=nonstopmode "test".tex
2 biber.exe "test"
3 xelatex.exe -no-pdf -interaction=nonstopmode "test".tex
4 xelatex.exe -synctex=1 -interaction=nonstopmode "test".tex
```

其中: -no-pdf 选项表示不生成 pdf 文档, -interaction=nonstopmode 选项表示出现错误不中断编译, -synctex=1 选项表示 pdf 文档与 tex 源文档进行跳转关联。

利用脚本编译原理是一样的, 只是利用编译脚本文件进行自动编译 (注意:linux 下的脚本文件需要添加一下路径)。比如:

示例 19 window 下的 bat 脚本文件

代码

```
1 @echo off
2 :: compile the tex file
3 xelatex.exe --synctex=-1 "test".tex
4 :: compile bibliography
5 biber "test"
6 :: compile again
7 xelatex.exe --synctex=-1 "test".tex
8 ::to do it again for backref
9 xelatex.exe --synctex=-1 "test".tex
10 :: clear aux files
11 del /q *.aux *.bbl *.blg *.log *.out *.toc *.bcf *.xml *.synctex *.nlo *.nls *.bak *.ind *.idx *.ilg *.lof
    *.lot *.ent-x *.tmp *.ltx *.los *.lol *.loc *.listing *.gz
```

示例 20 linux 下的 sh 脚本文件

代码

```
1 #!/bin/bash
2 # exec path for tex live 2016
3 export PATH=${PATH}:/usr/local/texlive/2016/bin/x86_64-linux
4
5 #compile the tex file
6 xelatex --synctex=-1 "test".tex
7 #compile bibliography
8 biber "test"
9 #compile again
10 xelatex --synctex=-1 "test".tex
11 #to do it again for backref
12 xelatex --synctex=-1 "test".tex
13 #clear aux files
14 rm -r *.aux *.bbl *.blg *.log *.out *.toc *.bcf *.xml *.synctex *.nlo *.nls *.bak *.ind *.idx *.ilg *.lof
    *.lot *.ent-x *.tmp *.ltx *.los *.lol *.loc *.listing *.gz
```

### 2.4.5 常见的编译出错问题

编译出现问题从多数读者的实践看主要包括 3 类:

一是编译命令使用不当。由于一些读者对传统基于 bibtex 的方法与基于 biblatex 的方法的差别并不了解, 以为编译的命令与传统方法一致, 会使用 bibtex 程序而不是 biber 程序对文献进行编译, 这就导致编译失败。这种情况下把编译命令改正确即可。(附带说明一下, 其

实基于 biblatex 的方法使用 bibtex 程序编译文献也是可以，把后端程序设置改为 bibtex 就可以，只是会失去很多功能特性，特别是对中文的一些支持会失效，所以并不提倡使用。而且要注意：使用 bibtex 后端，不是转换成传统的基于 bibtex 的方法，不能使用各种 bst 文件，换成 bibtex 后端仍然使用 cbx/bbx 样式)。

二是编译过程不完整。有时从以前编译过的文档中进行编译，可能由于 biblatex 升级等原因，原有的信息会导致出错。或者由于更换样式，而各个样式的输入输出不一样，导致编译失败。或者有时在 tex 文档中使用 filecontents 来写 bib 文件，但更改 bib 内容后，xelatex 编译不修改 bib 文件，导致没有新的文献条目而出错。这种情况下处理很简单，把所有的辅助文件清理干净，重新编译即可。

三是 biber 程序缓存被破坏。由于 biber 在第一次运行时会在缓存文件夹下创建一些二进制运行文件，如果这些文件被破坏，那么 biber 运行结果会出现莫名其妙的错误。比如：biber 运行一半就结束了，log 信息如下所示：

```
INFO - This is Biber 2.16
INFO - Logfile is 'thuthesis-example.blg'
INFO - Reading 'thuthesis-example.bcf'
INFO - Found 34 citekeys in bib section 0
INFO - Processing section 0
INFO - Looking for bibtex format file 'ref/refs.bib' for section 0
INFO - LaTeX decoding ...
INFO - Found BibTeX data source 'ref/refs.bib'
```

这次运行中，只是找到了 bib 文件，并没有下一步的处理，显然是存在问题的。

因此如果出现这种情况，可以手动删除缓存文件夹，让 biber 重新创建。这些缓存文件夹名字通常为：

```
par-<hex_encoded_username>/cache-*
par-<hex_encoded_username>\cache-*
```

在不同的系统下面，缓存文件夹的位置是不同的，可能的位置包括：

```
/var/folders/*/*/*/* (OSX, local GUI login shell)
/var/tmp/ (OSX (remote ssh login shell), Unix)
/tmp/ (Linux)
C:\Documents and Settings\<username>\Local Settings\Temp (Windows XP/Cyg- win)
C:\Windows\Temp (Windows)
C:\Users\<username>\AppData\Local\Temp (Windows 7/8)
```

也可以从运行得到的.blg 文件或者命令 `biber --cache` 来找到缓存文件夹。

删除缓存文件夹操作，在 Linux 和 Mac 下可以使用如下命令操作：`rm -rf `biber --cache``。

在 windows 下，那么其实随便用个清理软件把临时文件夹清理一下就行。直接点就是找到 `par-<hex_encoded_username>/cache-*` 缓存文件夹然后删除它。参考问题见 [biblatex-biber-fails-with-a-strange-error-about-missing-recode-data-xml-file](#)。

可能还有一些其它的不太常见的问题，比如 biblatex 更新失败，tex 编译环境安装不完整等，这些问题可以采用重装 texlive，重新更新 biblatex 等方法解决。

## 2.5 分章参考文献和书后参考文献

分章参考文献在书籍写作中是一种比较常见的需求。在 LaTeX 传统方法中, 利用 thebibliography 环境是可实现的, 但所有文献内容都需要手动输入, 且格式只能在 bibitem 内设置, 在文献量较大时显然是不适合的。

而利用 bibliographystyle 和 bibliography 命令生成的参考文献可以与 thebibliography 环境同时存在, 生成多个不同的文献表, 可以插入多个文献表但却是相同的, 这是因为编译文档时, 所有的信息都是写入一个 aux 文件中的。所以要实现分章参考文献表, 需要使用 chapterbib、multibib 等宏包, 使用 chapterbib 时要把需要生成文献表的章放到单独的 tex 子文档中, 然后用 include 命令包含进主文档, 这样可以生成多个 aux 文件以便生成分章参考文献表。其核心原理是利用多个 aux 文件区分不同部分的参考文献, bibunits、multibib 宏包尽管能在一个源文档 (非主文档和子文档的形式) 生成分章参考文献, 也是通过写多个 aux 文件实现的。

而 biblatex 宏包的机制则要灵活的多, 不再依赖多 aux 文件划分来实现文献的分章, 可以很方便的在一个 L<sup>A</sup>T<sub>E</sub>X 文档中实现多种形式的参考文献划分, 包括利用 refsection、refsegment 环境或者宏包选项等。

### 2.5.1 利用 refsection 环境分章

利用 refsection 环境可以显式的设置需要划分的参考文献的文档结构部分, 比如把 chapter 的所有内容放在 refsection 环境内, 然后在其中引用参考文献, 并使用参考文献打印命令 printbibliography 就会得到该环境内的参考文献表。例 21 给出的代码, 结果如图 11, 12 所示。

示例 21 分章参考文献

代码

```
1 \documentclass{report}%%file:egrefsection.tex
2 \usepackage{ctex}
3 \usepackage[backend=biber,style=gb7714-2015]{biblatex}
4 \addbibresource[location=local]{example.bib}
5
6 \begin{document}
7 \chapter{序章}
8 \begin{refsection}
9 序章内容\cite{GPS1988--}
10 \printbibliography[heading=subbibliography,title=本章参考文献]
11 \end{refsection}
12
13 \chapter{正文章一}
14 \begin{refsection}
15 正文内容一\cite{杨洪升2013-56-75}
16 \printbibliography[heading=subbibliography,title=本章参考文献]
17 \end{refsection}
18
19 \chapter{正文章二}
20 \begin{refsection}
21 正文内容二\cite{马克思2013-302-302}
22 \printbibliography[heading=subbibliography,title=本章参考文献]
23 \end{refsection}
24 \end{document}
```

### 2.5.2 利用宏包选项 refsection 分章

除了利用 refsection 环境显式方法外, 还可以利用宏包选项 refsection=chapter 等来设置需要划分的打印参考文献的文档结构, 这里设置 refsection=chapter 表示以章为单位区分打印

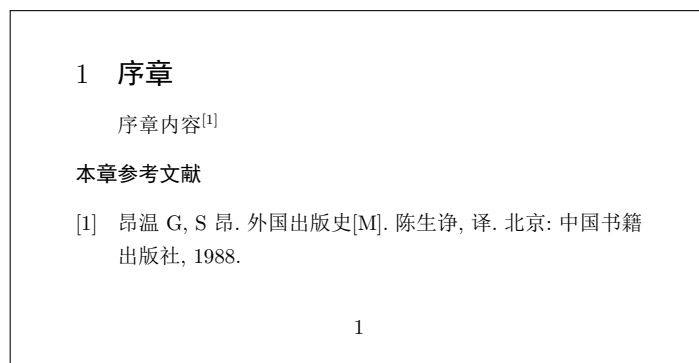


图 11 分章参考文献举例-序章

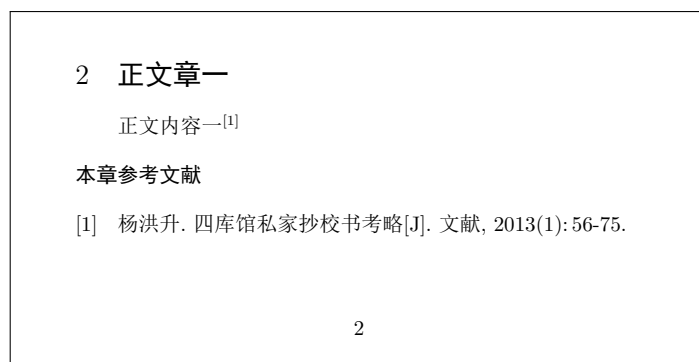


图 12 分章参考文献举例-章一

参考文献，也可以设置成 section，那么就是以节为单位进行区分打印。例22给出的代码，结果如图13所示，可以看到其结果与图11完全一致。

需要注意的是，bibtex 使用 refsection 选项 (而不是显式的 refsection 环境)，实质是对 chapter 或 section 等命令进行 patch 来进行文献分节处理，所以会对这些命令进行重定义，因此会导致与其它需重定义这些命令的宏包的冲突，比如 titlesec 宏包。当然这种冲突也只是在使用 refsection 选项的情况下，当不使用该选项时，bibtex 也不会做 patch，因此也就不会产生冲突。

#### 示例 22 利用宏包选项实现分章参考文献

代码

```

1 \documentclass{report}%%file:egrefsectionb.tex
2 \usepackage{ctex}
3 \usepackage[backend=biber,refsection=chapter,style=gb7714-2015]{bibtex}
4 \addbibresource[location=local]{example.bib}
5
6 \begin{document}
7 \chapter{序章}
8 序章内容\cite{GPS1988--}
9 \printbibliography[heading=subbibliography,title=本章参考文献]
10
11 \chapter{正文章一}
12 正文内容一\cite{杨洪升2013-56-75}
13 \printbibliography[heading=subbibliography,title=本章参考文献]
14
15 \chapter{正文章二}
16 正文内容二\cite{马克思2013-302-302}
17 \printbibliography[heading=subbibliography,title=本章参考文献]
18 \end{document}

```

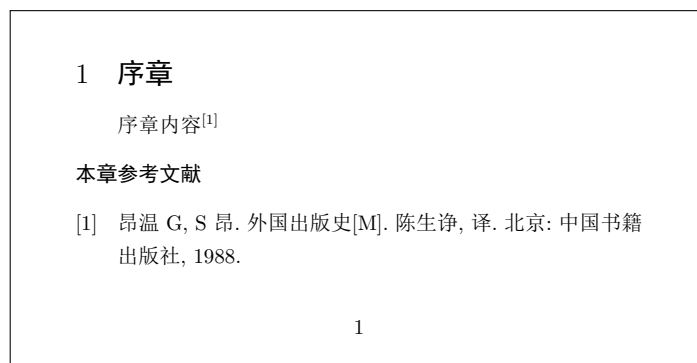


图 13 设置宏包选项实现分章参考文献

### 2.5.3 统一的全局参考文献

biblatex 利用 `refsection` 很容易生成分章节的参考文献。如果要生成一个全局的参考文献表, 那么可以不使用任何的 `refsection` 或者使用一个 `refsection` 包含全部的文档内容 (注意: 这两种方法一般情况下都是可行的, 但在 beamer 类第二种方法不能使用, 见第 2.13 节的说明)。

但还有一种需求是, 既要生成分章参考文献, 又要有一个全局的参考文献。那么最简单的方式是在文末利用 `printbibliography` 的 `section` 选项对所有的 `refsection` 同时打印一遍, 代码如例 23 所示, 但这样的方式更像是文献表堆积, 不像一个统一的文献表, 且其中的各章的文献引用标注序号不是全局的, 只是各 `refsection` 内部的序号。

示例 23 利用 `refsection` 的全局参考文献

代码

```
1 \printbibliography[section=1,heading=subbibliography,title=第一章参考文献]
2 \printbibliography[section=2,heading=subbibliography,title=第二章参考文献]
3 \printbibliography[section=3,heading=subbibliography,title=第三章参考文献]
4 \end{document}
```

这个问题可以在 `refsection` 外使用 `nocite` 后再打印一个全局文献表来解决, 也可以用 `refsegment` 代替 `refsection` 来解决。

`refsegment` 与 `refsection` 作用很像, 也用于划分, 但其内部的文献序号是全文统一的。每一个 `segment` 都有一个编号 (从 1 开始), 而在 `refsegment` 环境外的参考文献默认都在 `segment=0` 的 `segment` 内。用 `printbibliography` 命令打印在某个 `refsegment` 环境内的使用参考文献, 只需要给出 `segment=` 编号整数的一个值, 该值表示第几个 `refsegment` 环境。若 `printbibliography` 命令中给出 `segment=0` 则打印不在任一 `refsegment` 环境内的文献, 若不给出 `segment` 参数则遍历所有可以打印的参考文献, 包括各个 `refsegment` 内的文献。

注意: `refsection` 内部不可以有 `refsection`, 即 `refsection` 不可嵌套, 但其内部可以使用 `refsegment`。`printbibliography` 命令也可以对 `refsection` 内的所有 `refsegment` 遍历打印, 也可以指定 `refsection` 内的某一个 `refsegment` 中的文献, 只要给出 `segment` 参数, 当 `printbibliography` 在 `refsection` 外打印其内 `refsegment` 中的文献, 则还需指定 `section` 参数。

例 24 给出一个示例代码, 其结果如图 14 所示。其中“文献全局”命令遍历打印了所有不在 `refsection` 内的文献, 包括 `refSegment A`, `refSegment B`, 以及不在其内的 Gradshteyn 文献和附录的 Section E 中的 Parsons 文献。

注意到, 其中全局文献表的顺序编码是正常的全局顺序, 但对于每一分段 (`segment` 内) 该编码就不是当前段内的编码顺序。如果使用 `defernumbers` 选项, 那么可以将每个 `segment`



内的编码改为第一次打印该文献时的编码，但因为 segment 内的文献编码都是全局的，所以无法在一个 segment 内从 1 开始编码。

另外，如果不使用 refsegment 来实现全局的文献表，那么完全可以利用 nocite 命令在文档末尾引入需要出现在全局文献表中的文献，但此时文献表中文献的顺序可能与各章出现的顺序不一致，但从显示文献表的角度，这是完全没有问题的。

#### 示例 24 全局参考文献综合示例

代码

```
1 %file:egbibdiv.tex
2 %file:egbibdiva.tex
```

见：egbibdiv.tex，egbibdiva.tex。

参考文献测试 [2]。

第 1 节 refSegment A

分章节参考文献测试 [1]

文献 A

[1] M. Chiani, D. Dardari, and M. K. Simon. "New exponential bounds and approximations for the computation of error probability in fading channels". In: *IEEE Trans. Wireless Commun.* 2.4 (2003), pp. 840–845.

参考文献测试 [4]

第 2 节 refSegment B

文献 B

[4] 张敏莉, 易仕和, and 赵玉新. "超声速短化喷管的设计和试验研究". In: *空气动力学报* 25.4 (2007), pp. 500–503.

参考文献测试 [1]

第 3 节 refsection C

分章节参考文献测试 [1]

第 4 节 refsection D

§ 4.1 refsegment D-1

分章节参考文献测试 [3]。

§ 4.2 refsegment D-2

分章节参考文献测试 [2]。

文献 C

[1] Min-li Zhang, Shi-he Yi, and Yu-xin Zhao. "The design and experimental investigations of supersonic length shorted nozzle". In: *ACTA AERODYNAMICA SINICA* 25.4 (2007), pp. 500–503.

文献 D

[1] J. B. Andersen, T. S. Rappaport, and S. Yoshida. "Propagation measurements and models for wireless communications channels". In: *IEEE Commun. Mag.* 33.1 (1995), pp. 42–49.

[2] S. Lin and J. Daniel Costello. *Error Control Coding*. Second Edition. Englewood Cliffs: NJ:Prentice Hall, 2004.

[3] M. K. Simon and M. S. Alouini. *Digital Communication over Fading Channels*. Hoboken: NJ: Wiley-IEEE Press, 2004.

文献全局

[1] M. Chiani, D. Dardari, and M. K. Simon. "New exponential bounds and approximations for the computation of error probability in fading channels". In: *IEEE Trans. Wireless Commun.* 2.4 (2003), pp. 840–845.

[2] I. S. Gradshteyn and I. M. Ryzhik. *Tables of Integrals, Series, and Products*. Sixth Edition. San Diego: CA: Academic Press, Inc., 2000.

[3] J. D. Parsons. *The Mobile Radio Propagation Channel*. 2nd ed. Hoboken: John Wiley and Sons, 2000.

[4] 张敏莉, 易仕和, and 赵玉新. "超声速短化喷管的设计和试验研究". In: *空气动力学报* 25.4 (2007), pp. 500–503.

第 A 节 Section E

参考文献测试 [3]。

1

图 14 全局参考文献综合示例

#### 2.5.4 正文和附录区分的参考文献表

有时一些学位论文会要求区分正文和附录中的参考文献，也就是要生成不同的文献表。若只是两个常规的文献表，那么是容易实现的，只要使用两个 refsection 就可以。

但很多时候附录文献表对于文献条目的顺序数字标签是有要求的, 不如附录 A 中, 要求标签为 A1,A2 这样的形式, 而附录 B 中, 则要求是 B1,B2 这样的形式, 这种情况下需要特殊的处理。

这时要使用 `refcontext`。`refcontext` 可以用在临时调整参考文献的排序等方面, 也可以用于给文献条目的标签加上前缀 (prefix)。

我们用一个例子来表示:

示例 25 区别于正文的附录带前缀参考文献示例

代码

```

1 \documentclass{ctexbook}
2 \usepackage[paperheight=8cm,vmargin=0.5cm]{geometry}
3 \usepackage{xcolor}
4 \usepackage[colorlinks]{hyperref}
5 \usepackage[backend=biber,style=gb7714-2015]{biblatex}%sorting=nyt
6
7 \begin{filecontents}[force]{\jobname.bib}
8 @Book{Peebles2001-100-100,
9   Title = {Probability, random variable, and random signal Principles and \LaTeX{}},
10  Address = {New York},
11  Author = {von Peebles, Jr., P. Z.},
12  Edition = {4},
13  Pages = {100},
14  Publisher = {McGraw-Hill},
15  Year = {2001}
16 }
17
18 @Book{蔡敏2006--,
19   Title = {UML基础和Rose建模教程},
20   Address = {北京},
21   Author = {蔡敏 and 徐慧慧 and 黄柄强},
22   Publisher = {人民邮电出版社},
23   Year = {2006},
24   Month = {1}
25 }
26 \end{filecontents}
27 \addbibresource{\jobname.bib}
28 \DeclareRefcontext{ap}{labelprefix=A}
29 \let\cleardoublepage\clearpage
30
31 \begin{document}
32 \chapter{正文}
33 \begin{refsection}
34 正文内容
35 \cite{Peebles2001-100-100}
36 \cite[49]{蔡敏2006--}
37 \printbibliography
38 \end{refsection}
39
40 \appendix
41 \chapter{附录一}
42 \newrefcontext{ap}
43 \begin{refsection}
44 附录内容
45 \cite{Peebles2001-100-100}
46 \cite[49]{蔡敏2006--}
47 \printbibliography[heading=subbibliography]
48 \end{refsection}

```

```

49
50 \chapter{附录二}
51 \newrefcontext[labelprefix=B]
52 \begin{refsection}
53 附录内容
54 \cite{Peebles2001-100-100}
55 \cite[49]{蔡敏2006--}
56 \printbibliography[heading=subbibliography]
57 \end{refsection}
58 \end{document}

```

这样会形成 3 个文献表，一个正文的，两个则分别是附录一和二。

上述各个附录中采用单独的文献表，所以使用 `refcontext` 时，标签是不容易冲突的，但若要求附录要使用一个全局的文献表，又要求各个附录中文献的标签前缀不一样，此时直接如上面例子的方式使用 `refcontext` 是不行的。此时还要配合 `refsection` 或者 `refsegment` 来实现。下面给出例子：

示例 26 各附录不同前缀的附录单文献表示例

代码

```

1 \documentclass{article}
2 \begin{filecontents}[force]{\jobname.bib}
3 @article{A,title={A}}
4 @article{B,title={B}}
5 @article{C,title={C}}
6 @article{D,title={D}}
7 @article{E,title={E}}
8 @article{F,title={F}}
9 \end{filecontents}
10 \usepackage[style=gb7714-2015,defernumbers=true]{biblatex}%
11 \addbibresource{\jobname}
12
13 \begin{document}
14 \section{mainbody}
15 text\cite{A}
16 text\cite{B}
17 \printbibliography
18
19 \appendix
20 \begin{refsection}
21 \begin{refsegment}
22 \section{appendix 1}
23 text\cite{C}
24 text\cite{D}
25 \end{refsegment}
26
27 \begin{refsegment}
28 \section{appendix 2}
29 text\cite{E,F}
30 \end{refsegment}
31
32 \newrefcontext[labelprefix={A.}]
33 \printbibliography[segment=1]
34 \vspace{-1.3\baselineskip}
35 \newrefcontext[labelprefix={B.}]
36 \printbibliography[heading=none,segment=2]
37 \end{refsection}
38 \end{document}

```

```

39
40 %%%%%%%%%%或者
41
42 \documentclass{article}
43 \begin{filecontents}[force]{\jobname.bib}
44 @article{A,title={A}}
45 @article{B,title={B}}
46 @article{C,title={C}}
47 @article{D,title={D}}
48 @article{E,title={E}}
49 @article{F,title={F}}
50 \end{filecontents}
51 \usepackage[style=gb7714-2015]{biblatex} %,defernumbers=true
52 \addbibresource{\jobname}
53
54 \begin{document}
55 \section{mainbody}
56 text\cite{A}
57 text\cite{B}
58 \printbibliography
59
60 \appendix
61 \begin{refsection}
62 \section{appendix 1}
63 text\cite{C}
64 text\cite{D}
65 \end{refsection}
66
67 \begin{refsection}
68 \section{appendix 2}
69 text\cite{E,F}
70 \end{refsection}
71
72 \newrefcontext[labelprefix={A.}]
73 \printbibliography[section=1]
74 \newrefcontext[labelprefix={B.}]
75 \vspace{-1.3\baselineskip}
76 \printbibliography[heading=none,section=2]
77 \end{document}

```

## 2.6 参考文献著录和标注样式

参考文献著录和标注样式是排版的主要内容之一，但对于普通用户来说只要学会选择使用即可，因为其格式已经由样式所设定。当然如果用户需要在某种样式的基础上有进一步的修改，那么完全可以开展这方面的工作。其中著录样式的在 \*.bbx 文件中修改，而标注样式在 \*.cbx 文件中修改。

### 2.6.1 标准样式

biblatex 附带了一些标准的样式，比如 numeric, authoryear，这两个样式是比较常用的，也是样式作者定制自定义样式的重要基础。其它著录样式，比如 alphabetic, authortitle, verbose 等也可在一些场合使用 (详见 biblatex 手册 Bibliography Styles 一节)。标注样式除了 numeric, authoryear 外，也还有 numeric-comp, numeric-verb, alphabetic, authortitle 等 (详见 biblatex 手册 Citation Styles 一节)。

特别需要强调的是：*numeric-comp*, *authoryear-comp* 是针对顺序编码和作者年制的两个标签压缩样式，顺序编码制的压缩样式主要针对多篇文献传递给一个标注命令的情况，而作者

年制的压缩样式除了给一个标注命令传递多篇文献外，还要求连续引用的文献具有相同作者，而要使相同作者的文献连续，可以通过手动方式给出，也可以通过自动排序方式实现。

读者可以参考 biblatex 附带的 96 个示例文件，也可以测试这些样式的效果，以便在需要的时候选择使用。而且著录和标注可以分别使用不同的样式，著录是一种，而标注则是另一种，实际使用的例子可以参考：[参考文献的顺序](#)。

## 2.6.2 其它常用样式

除标准样式外，在 texlive 安装目录中或者 ctan 上面可以找到更多的常用样式，比如国外的 APA、MLA 样式，国内的 GB/T 7714-2015 等标准的样式，也有一些针对某些期刊的文献样式，比如：nature, science, nejm 等等，也有一些是第三方作者所做的具有自己风格的样式，比如：biblatex-casperiector 样式包。用户可以测试各样式包，并按需使用，进一步也可以参考其代码，修改自己需要的样式。表3列出了这些常用的样式。

表 3 常用的 biblatex 样式

样式名	对应的 bibtex 样式	作者介绍	样式说明
trad-plain	plain	MarcoDaniel and MoritzWemheuer, 后者是 biblatex 维护者之一	将引文按字母顺序排序, 比较次序为作者姓氏、出版年份和题名, 如果不能顺序, 将以在正文中的引用顺序为准。
trad-unsrt	unsrt	MarcoDaniel and MoritzWemheuer	按照在正文中引用文献的先后顺序排列文献, 其排版格式与 trad-plain 基本相同
trad-alpha	alpha	MarcoDaniel and MoritzWemheuer	用文献的作者姓氏前三个字母加出版年份的后两位数作为文献序号, 如果出现相同的序号, 则会根据排序结果在序号后追加字母以示区别, 排序方法和排版格式与 trad-plain 相同
trad-abbrev	abbrev	MarcoDaniel and MoritzWemheuer	将文献中作者名和月份名的拼写改为缩写, 显得文献信息紧凑简洁, 其排序方法和排版格式与 trad-plain 相同
ieee	IEEEtran	Joseph Wright, biblatex 维护者之一	国际电气电子工程师协会 IEEE 期刊文献格式
apa	apalike	Philip Kime, biblatex 作者之一	American Psychological Association 的文献格式
Chicago	Chicago	David Fussner	for the Chicago Manual of Style
iso-numeric		Michal Hoftich	ISO690 international standard numeric system
iso-iso-authoryear		Michal Hoftich	ISO690 international standard nameanddate system,so-called Harvard style
gb7714-2015	gbt7714-numeric	hushidong	中文文献著录标准 GB/T 7714-2015 顺序编码制
gb7714-2015ay	gbt7714-authoryear	hushidong	中文文献著录标准 GB/T 7714-2015 著者年份制
casperiector		Casper vector	一种中文文献格式
nature		Joseph Wright	for Nature
science		Joseph Wright	for Science
chem-acsc		Joseph Wright	covers most American Chemistry Society journals
chem-angew		Joseph Wright	covers Angewandte Chemie Chemistry-A European Journal.
chem-biochem		Joseph Wright	covers Biochemistry and a small number of other American Chemistry Society journals
chem-rsc		Joseph Wright	covers all Royal Society of Chemistry journals
phys		Joseph Wright	for AIP and APS
nejm		MarcoDaniel	for New England Journal of Medicine
mla		James Clawson	for Modern Language Association
author-title-dw		Dominik Waßenhoven	for Humanities
footnote-dw		Dominik Waßenhoven	for Humanities



### 2.6.3 biblatex-gb7714-2015 样式包

目前符合 GB/T 7714-2015 标准的 biblatex 样式, 是本文作者开发的 biblatex-gb7714-2015 样式包, 分顺序制和作者年制两类。其加载方式见例8。

该样式包, 实现了 GB/T 7714-2015 标准的完整要求, 包括两种编制方式下的各类型参考文献著录格式和标注格式等基本内容, 还包括: 自动判断文献类型和载体、自动并判断区分文献语言、双语对照文献, 带页码的标注格式, 作者年制下文献的自动排序和仅有年的标注格式, 两种编码制方式下责任者缺省不同处理, 其他信息缺省时的自动处理, 一些信息如页码卷期自动解析等功能。

该样式包增加了标签的对齐方式 (gbaligned 选项)、出版项缺省时的处理 (gbpub 选项)、作者缺省时的处理 (gbnoauthor 选项)、作者姓名格式的调整 (gbnamefmt 选项)、文献类型和载体标识的输出控制 (gbtype 选项)、析出文献标识符//控制 (gbpunctin 选项)、GBK 编码文档兼容性 (gbcodegbk 选项)、域的默认格式还原 (gbfieldstd 选项)、标题的超链接控制 (gbtitlelink 选项) 等。能通过标注和著录分别采用著者-出版年制 (即作者年制)、顺序编码制样式实现特殊的参考文献格式, 并尽最大可能利用 biblatex 的选项来实现一些特殊需求, 比如文献表中作者数量大于 3、标注中仅给出第一作者的姓、参考文献表按拼音或笔画排序等等。

该样式包可与 biblatex v2.8 (texlive2014 发行版附带) 以上所有版本配合使用 (注意 ctex 2.9.4 的用户若想使用需升级 biblatex)。兼容传统的参考文献条目类型, 比如 www/electronic/conference/mastersthesis/phdthesis/techreport/standard, 并根据国标要求增加了 newspaper(报纸析出的文献)、database(数据库)、dataset(数据集)、software(软件)、map(舆图)、archive(档案) 等类型。兼容适用于不同样式的 bib 数据源, 可与 gbt7714 宏包的 bst 样式使用的 bib 文件无缝衔接。

关于参考文献数据录入准备、条目类型的域构成、样式包使用说明等更多的内容详见<sup>[10]</sup>。在使用时还需要注意一种情况, 即在一部作品中, 同时存在上标标注和非上标标注, 在标准样式下有 supercite 命令提供上标模式, 而 cite 和 parencite 命令提供非上标模式。而在 gb7714-2015 样式中, supercite 和 cite 命令提供上标模式, 而 parencite 命令提供非上标模式。

关于该样式包的安装, 在 2017 年以后 tex 发行版本中, 不再需要安装, 可直接使用, 因为 texlive 和 miktex 均已收录。而老版本的 texlive 比如 texlive 2016 及以下版本下可以采用离线方式安装, 步骤如下:

(a) 在 ctan 上面下载 biblatex-gb7714-2015 宏包, 下载地址:<https://www.ctan.org/pkg/biblatex-gb7714-2015>

(b) 解压压缩包, 放到:texlive 的 texmf-dist/tex/latex 或者 texmf-local/tex/latex 或者它们的子目录下面。(参考: 文献 [11] 的 3.4.6 节)

(c) 运行 texhash 或者 mktexlsr 命令, 刷新文件名数据库。

关于该样式包的升级, 一般情况下可利用发行版的升级工具升级。比如使用最新的 texlive 发行版, 那么可以使用 tlmgr 更新宏包, 步骤如下:

(a) 打开 tlmgr-gui, 在选项通用里面设置国内的 ctan 镜像源, 比如<http://mirrors.ustc.edu.cn/CTAN/systems/texlive/tlnet>

(b) 点击加载缺省软件包仓库 (注意加载完成后可能要求升级 tlmgr, 那么可先升级 tlmgr)。

(c) 在完成加载后, 在匹配文本框填入 biblatex-gb7714-2015, 搜索更新即可。

因为 texlive 每年更新一次, 老的版本则会冻结, 所以一般情况下, 老版本 texlive 中宏包是无法通过 tlmgr 升级到最新的版本的。因此通常的方式是下载新的宏包, 用离线下载的方式覆盖安装, 即将下载的 cbx/bbx 文件覆盖系统中 tex 发行版内对应的 cbx/bbx 文件。

## 2.6.4 标注命令

传统方法中类似 natbib, mcite 这样的宏包提供了很多有特色的标注命令。比如:

示例 27 natbib 和 mciteplus 等宏包提供的标注名

代码

```

1 \citep{jon90} fora parenthetical citation(Jones et al.,1990),
2 \citet{jon90} fora textual one, as Jones et al.(1990).
3 \citealt{jon90} Jonesetal.1990
4 \citealt*{jon90} Jones,Baker,andWilliams1990
5 \citealp{jon90} Jones et al.,1990
6 \citealp*{jon90} Jones,Baker,and Williams,1990
7 \citealp{jon90,jam91} Jones et al.,1990;James et al.,1991
8 \citealp[pg.~32]{jon90} Jones et al.,1990,pg.32
9 \citenum{jon90} 11
10 \citetext{priv.\comm.} (priv.comm.)
11
12 \citeauthor{jon90} ) Jonesetal.
13 \citeauthor*{jon90} ) Jones,Baker,andWilliams
14 \citeyear{jon90} ) 1990
15 \citeyearpar{jon90} ) (1990)
16
17 \cite{Glashow,*Salam,*Weinberg,Smith,*Jones}%multiplegroupsOK
18 \cite{Glashow,*Salam,*Weinberg}\cite{Glashow}%headcanberecitedanytime
19 \cite{Glashow,*Salam,*Weinberg}\cite{Glashow,*Salam,*Weinberg}%restatepreviousdefinition
20 \cite{Glashow,*Salam,*Weinberg}\cite{Glashow,*Weinberg}%restatepartofpreviousdefinition
21 \cite[page580ofGlashow]{Glashow,*Salam}%optionalargumentissupported
22 \nocite{*}%BibTeXwildcardOK,butusewithcaution

```

biblatex 也提供功能类似、语法相似的命令 (或直接给出, 或通过相应的模块给出)。除此之外, biblatex 还提供具有特点的更多标注命令, 以及强大的标注命令定制功能。而这些标注命令常常是与样式相关的, 因为这些样式通常会重定义标注命令。biblatex-gb7714-2015 中的常用标注命令, 如例28所示, 更多命令及其说明详见 biblatex-gb7714-2015.pdf。

示例 28 常见的标注命令

代码

```

1 1. 不带页码的引用(上标, 方括号包围):
2 \cite{Peebles2001-100-100}
3
4 2. 不带页码的引用(非上标, 方括号包围):
5 \parencite{Miroslav2004--}
6
7 3. 带页码的引用:
8 \cite[见][49页]{蔡敏2006--} \parencite[见][49页]{Miroslav2004--}
9 \pagescite{Peebles2001-100-100} \pagescite[][201-301]{Peebles2001-100-100}
10
11 4. 顺序编码制中同时输出作者和顺序编码标签, 比如:
12 (一) 直接的方法: 见\citeauthor{refb}\cite{refb},\citeauthor{refc}\cite{refc}
13 (二) 定义新的标注命令: 见\authornumcite{refb,refc}
14 (三) 用textcite但没有上标: 见\textcite{refb,refc}
15
16 5. 作者年制文中已有作者还需要年份和页码的情况, 使用命令yearpagescite, 比如:
17 见赵耀东\yearpagescite[][205]{赵耀东1998--}和Simon\yearpagescite[][15]{Simon2001--}
18
19 6. 作者年制文中已有作者只需要年份的情况, 使用命令yearcite和手动方式, 比如:
20 见赵耀东\yearcite{赵耀东1998--}
21 见赵耀东(\cite*{赵耀东1998--})
22 见赵耀东(\citeyear{赵耀东1998--})

```

23

24 7. 在页脚中引用和打印文献表:

25 `\footnote{在脚注中引用\footcite{赵学功2001--}} \footfullcite{赵学功2001--}`

### 2.6.5 参考文献文境

参考文献文境是 biblatex 高度可定制性可扩展性的体现。biblatex 认为文献的引用和打印都是在一个文境中的, 因此在不同的文境中就可以有不同的设置, 这些设置包括: 排序模板、构建姓名排序关键字的模板、使用字母或数值标签的标注格式的前缀字符串、计算姓名唯一性 (非歧义性) 信息的模板、构造姓名的字母顺序标签的模板。因此要在一个文档中使用不同的这些设置, 那么可以利用参考文献文境。文境除了用于设置影响文献表打印的选项, 设置的选项还可以影响标注命令输出的数据。biblatex 手册中给出了几个典型的应用场景, 包括: (1) 打印多个文献表, 不同的文献表使用不同的排序。(2) 打印多个文献表, 不同的文献表使用不同的排序模板, 进而导致排序使用的数据也是不同的。(3) 打印多个文献表, 不同的文献表使用不同的排序, 不同的文献表使用不同的标签。

另外一个应用场景是第 2.10.7 节提到的示例, 标注和文献表使用不同的排序。比如要求标注排序使用 `ynt`, 而文献表排序使用 `nyt`。

需要注意: 默认情况下, 用于标注的数据来自于打印该条目的最后一个参考文献表所在的著录文境。对于某些没有在文献表中打印条目但又以某种方式给出的标注命令, 它们默认从文档开始时建立的全局文境获取数据。要覆盖这一行为, 可以手动将标注命令放在 `refcontext` 环境内, 但是这样容易出错并且很繁琐。而 `assignrefcontext` 这些命令可以用来设置一个由 `keywords`、`categories` 或 `entrykeys` 构成的逗号分隔列表, 使带有这些指定关键词、或属于任何指定类别、或指定引用关键词的条目可以从 `refcontext key/values` 指出的特定文境获取数据, 这些 `refcontext key/values` 选项会被解析为对应的 `refcontext` 选项。

## 2.7 参考文献标题格式

期刊文章经常需要设置某种格式的参考文献标题, 包括字号, 大小, 段落格式等。在书籍等写作中有时还需要将参考文献加入目录中, 以便实现超链接。

### 2.7.1 加入目录超链接

加入目录的命令是 `addcontentsline`, 在 `printbibliography` 命令前使用该命令即可将参考文献加入目录中, 同时为了能够超链接正确可以加入 `hyperref` 宏包提供的 `phantomsection` 命令。例 29 给出了代码。

示例 29 手动加入目录链接

代码

```
1 \addcontentsline{toc}{chapter}{参考文献}
2 \phantomsection
3 \printbibliography[heading=bibliography,title=本章参考文献]
```

除了上述手动添加的方式外, 还可以在 `printbibliography` 命令使用 biblatex 提供的选项来实现, 比如 `heading=bibintoc`, 该选项与 `heading=bibliography` 是类似的, 只是增加了在目录中加入链接的功能, 例 30 给出了代码。

示例 30 使用 `bibintoc` 加入目录链接

代码

```
1 \printbibliography[heading=bibintoc,title=本章参考文献]
```

### 2.7.2 重定义 heading

默认情况下, biblatex 参考文献标题的格式主要依赖于 L<sup>A</sup>T<sub>E</sub>X 文档的章节的标题格式。当 printbibliography 中使用 heading 选项的参数是 bibliography 和 bibintoc 时, 参考文献标题的格式与当前文档类的主划分单元的标题格式一致, 比如 book 和 report 类中与 chapter 一致, 而在 article 类中与 section 一致。heading 选项的参数是 subbibliography 和 subbibintoc 时, 则与主单元的下一层级的标题格式一致, 即 book 和 report 类中与 section 一致, 而在 article 类中与 subsection 一致。因此设置参考文献标题的格式最简单方式是设置文档的各级标题格式然后选择 heading 选项的参数 (注意当 heading 选项不给出时, 其默认参数为 bibliography)。

事实上利用 defbibheading 命令重定义 bibliography 等 heading 的相关信息可以改变默认的标题对应方式, 比如可以将 bibliography 与 subsubsection 层级标题格式对应起来, 例31给出了代码。其中还使用了 phantomsection 和 addcontentsline 命令可以完成上一小节需要的加入目录链接功能。其中还使用了 centering 使标题居中, 这预示了一种标题格式修改方式, 甚至可以不使用文档类的标题格式而直接自定义标题。

示例 31 biblatex 对于目录的影响

代码

```
1 \usepackage[backend=biber,style=gb7714-2015ay]{biblatex}
2 \defbibheading{bibliography}{\bibname}{%
3 \phantomsection%解决链接指引出错的问题, 相当于加入了一个引导点
4 \addcontentsline{toc}{subsubsection}{#1}
5 \centering\subsubsection*{#1}}%
```

需要注意, 如果要使用 titleps 作为页眉页脚的设置宏包, 那么需要利用 defbibheading 重设一下 heading 选项参数, 因为默认情况下 heading 的参数比如 bibliography 中带有 markboth 命令, 该命令与 titleps 共用时会导致页眉页脚一些出现问题。例32给出了代码。而使用文档类默认的页眉页脚或者利用 fancyhdr 设置页眉页脚, 则不需要修改。为了方便移植应用, 这里给出三种设置页眉页脚的代码, 例33是使用文档类提供和 ctex 修改的页眉页脚; 例34是使用 fancyhdr 设置的页眉页脚; 例35是使用 titleps 设置的页眉页脚。

示例 32 重设 heading 选项参数

代码

```
1 \defbibheading{bibliography}{\bibname}{%
2 \chapter*{#1}}%
3 % \markboth{#1}{#1}
4 %重定义命令中去掉了markboth那一句命令。
```

示例 33 文档类提供 ctex 修改的页眉页脚举例

代码

```
1 %file:egheadclass.tex
2 \documentclass{book}
3 \usepackage{ctex}
4 \usepackage[paperwidth=12cm,paperheight=6cm,top=1.5cm, bottom=1.5cm, left=1cm,right=1cm]{geometry}
5 \usepackage{titlesec}
6 \titleformat{\chapter}{\zihao{4}\heiti}{\thechapter}{1em}{}
7 \titlespacing*{\chapter}{0pt}{0.0\baselineskip}{0.5\baselineskip}{0pt}
8 \titleformat{\section}{\zihao{5}\heiti}{\thesection}{1em}{}
9 \titlespacing*{\section}{0pt}{0.5\baselineskip}{0.5\baselineskip}{0pt}
10 \usepackage[backend=biber,style=nature]{biblatex}
11 \addbibresource[location=local]{example.bib}
12
13 \renewcommand{\bibfont}{\zihao{-6}\songti}
```

---

```

14 \setlength{\bibitemsep}{2pt}
15 \usepackage{hyperref}
16 \begin{document}
17 \tableofcontents
18 \begin{refsection}
19 \chapter{序章}
20 \section{序节}
21 序章内容\cite{GPS1988--}
22 \def\bibentryset{bilangyi2013}{易仕和2013--,Yi2013--}
23 专著, 双语文献引用\cite{bilangyi2013}
24
25 \phantomsection
26 \addcontentsline{toc}{chapter}{参考文献}
27 \printbibliography[heading=bibliography,title=本章参考文献]
28 \end{refsection}
29
30 \begin{refsection}
31 \chapter{正文一}
32 \section{正文节一}
33 正文内容一\cite{杨洪升2013-56-75}
34
35 \phantomsection
36 \addcontentsline{toc}{section}{参考文献}
37 \printbibliography[heading=subbibliography,title=本章参考文献]
38 \end{refsection}
39
40 \begin{refsection}
41 \chapter{正文二}
42 \section{正文节二}
43
44 正文内容二\cite{马克思2013-302-302}
45 \printbibliography[heading=subbibliography,title=本章参考文献]
46 \end{refsection}
47 \end{document}

```

---

### 示例 34 利用 fancyhdr 生成页眉页脚举例

代码

---

```

1 %file:egheadfancy.tex
2 \documentclass{book}
3 \usepackage{ctex}
4 \usepackage[paperwidth=12cm,paperheight=6cm,top=1.5cm, bottom=1.5cm, left=1cm,right=1cm]{geometry}
5 \usepackage{titlesec}
6 \titleformat{\chapter}{\zihao{4}\heiti}{\thechapter}{1em}{}
7 \titlespacing*{\chapter}{0pt}{0.0\baselineskip}{0.5\baselineskip}{0pt}
8 \titleformat{\section}{\zihao{5}\heiti}{\thesection}{1em}{}
9 \titlespacing*{\section}{0pt}{0.5\baselineskip}{0.5\baselineskip}{0pt}
10 \usepackage[backend=biber,style=nature]{biblatex}
11 \addbibresource[location=local]{example.bib}
12
13 \renewcommand{\bibfont}{\zihao{-6}\songti}
14 \setlength{\bibitemsep}{2pt}
15 \usepackage{hyperref}
16 \usepackage{fancyhdr}
17 \pagestyle{fancy}
18 \renewcommand{\chaptermark}[1]{%
19 \markboth{#1}{}
20 \renewcommand{\sectionmark}[1]{%
21 \markright{\thesection\ #1}}

```

```

22 \fancyhf{} % delete current header and footer
23 \fancyhead[LE,RO]{\bfseries\thepage}
24 \fancyhead[LO]{\bfseries\rightmark}
25 \fancyhead[RE]{\bfseries\leftmark}
26 \renewcommand{\headrulewidth}{0.5pt}
27 \renewcommand{\footrulewidth}{0pt}
28 \addtolength{\headheight}{0.5pt} % space for the rule
29 \fancypagestyle{plain}{%
30 \fancyhead{} % get rid of headers on plain pages
31 \renewcommand{\headrulewidth}{0pt} % and the line
32 }
33
34 \begin{document}
35 \tableofcontents
36 \begin{refsection}
37 \chapter{序章}
38 \section{序节}
39 序章内容\cite{GPS1988--}
40 \defbibentryset{bilangyi2013}{易仕和2013--,Yi2013--}
41 专著, 双语文献引用\cite{bilangyi2013}
42
43 \phantomsection
44 \addcontentsline{toc}{chapter}{参考文献}
45 \printbibliography[heading=bibliography,title=本章参考文献]
46 \end{refsection}
47
48 \begin{refsection}
49 \chapter{正文一}
50 \section{正文节一}
51 正文内容一\cite{杨洪升2013-56-75}
52
53 \phantomsection
54 \addcontentsline{toc}{section}{参考文献}
55 \printbibliography[heading=subbibliography,title=本章参考文献]
56 \end{refsection}
57
58 \begin{refsection}
59 \chapter{正文二}
60 \section{正文节二}
61 正文内容二\cite{马克思2013-302-302}
62 \printbibliography[heading=subbibliography,title=本章参考文献]
63 \end{refsection}
64 \end{document}

```

### 示例 35 利用 titles 生成页眉页脚举例

代码

```

1 %file:egheadtitles.tex
2 \documentclass{book}
3 \usepackage{ctex}
4 \usepackage[paperwidth=12cm,paperheight=6cm,top=1.5cm, bottom=1.5cm, left=1cm,right=1cm]{geometry}
5 \usepackage[pagestyles]{titlesec}
6 \titleformat{\chapter}{\zihao{4}\heiti}{\thechapter}{1em}{}
7 \titlespacing*{\chapter}{0pt}{-1.0\baselineskip}{0.5\baselineskip}{0pt}
8 \titleformat{\section}{\zihao{5}\heiti}{\thesection}{1em}{}
9 \titlespacing*{\section}{0pt}{0.5\baselineskip}{0.5\baselineskip}{0pt}
10 \usepackage[backend=biber,style=nature]{biblatex}
11 \addbibresource[location=local]{example.bib}
12

```



```

13 \renewcommand{\bibfont}{\zihao{-6}\songti}
14 \setlength{\bibitemsep}{2pt}
15 \defbibheading{bibliography}[\bibname]{%
16 \phantomsection%
17 \addcontentsline{toc}{chapter}{#1}%
18 \chapter*{#1}}%
19 \defbibheading{subbibliography}[\bibname]{%
20 \phantomsection%
21 \addcontentsline{toc}{section}{#1}%
22 \section*{#1}}%
23 \usepackage{hyperref}
24
25 \newpagestyle{main}{%偶数页
26 \sethead[\small$\cdot$~\thepage~$\cdot$] []
27 [\small\,\,thesection\quad\sectiontitle]%奇数页
28 {\small\,\,thethechapter\quad\chaptertitle}{\small$\cdot$~\thepage~$\cdot$}
29 \setfoot{}{}{}\headrule\footrule}
30 %注意\sectiontitle应该是titlesec宏包定义的命令
31 \renewcommand\headrule{\setheadrule{1pt}}
32 \renewcommand\footrule{\setfootrule{0pt}}
33
34 \newpagestyle{premain}{
35 \sethead[\small$\cdot$~\thepage~$\cdot$] [] [\small\,\,目录]
36 {\small\,\,leftmark}{\small$\cdot$~\thepage~$\cdot$}
37 \setfoot{}{}{}\headrule\footrule}
38
39 \newpagestyle{pgref}{%
40 \sethead[\small\,\,~\thepage~]% 偶数页左
41 []% 偶数页中
42 [\small\,\,参考文献]% 偶数页右
43 {\small\,\,thethechapter\quad\chaptertitle\hfil}% 奇数页左
44 {}% 奇数页中
45 {\small\,\,~\thepage~}% 奇数页右
46 \setfoot{}{}{}%
47 \headrule%
48 }%
49
50 \begin{document}
51 \pagestyle{premain}
52 \tableofcontents
53 \cleardoublepage
54
55 \begin{refsection}
56 \pagestyle{main}
57 \chapter{序章}
58 \section{序节}
59 序章内容\cite{GPS1988--}
60 \defbibentryset{bilangyi2013}{易仕和2013--,Yi2013--}
61 专著，双语文献引用\cite{bilangyi2013}
62 %\newpage
63 %\vfil\hspace{1pt}
64 %\newpage
65 \pagestyle{pgref}
66 \printbibliography[heading=bibliography,title=本章参考文献]
67 \cleardoublepage%页眉页脚分割正确需要该命令
68 \end{refsection}
69
70 \begin{refsection}

```

```

71 \pagestyle{main}
72 \chapter{正文 章一}
73 \section{正文 节一}
74 正文内容一\cite{杨洪升2013-56-75}
75 %\newpage
76 %\vfil\hspace{1pt}
77 %\newpage
78 \pagestyle{pgref}
79 \printbibliography[heading=subbibliography,title=本章参考文献]
80 \cleardoublepage
81 \end{refsection}
82
83 \begin{refsection}
84 \pagestyle{main}
85 \chapter{正文 章二}
86 \section{正文 节二}
87 正文内容二\cite{马克思2013-302-302}
88 %\newpage
89 %\vfil\hspace{1pt}
90 %\newpage
91 \pagestyle{pgref}
92 \printbibliography[heading=subbibliography,title=本章参考文献]
93 \cleardoublepage
94 \end{refsection}
95 \end{document}

```

### 2.7.3 利用 titlesec

既然参考文献的标题可以使用文档类的标题格式,且 titlesec 的标题样式命令具有局部性,那么完全可以利用 titlesec 宏包定义一个需要的标题格式,在参考文献表中使用,使用完以后再恢复原来的设置。(注意:当使用 titlesec 宏包重定义 section 等标题样式后,在 defbibheading 命令中使用 centering 可能无效)。例36给出了测试代码,其结果如图15所示。

示例 36 利用 titlesec 局部修改参考文献标题格式

代码

```

1 %file:egsectitle.tex
2 \documentclass{book}
3 \usepackage{ctex}
4 \usepackage[paperwidth=12cm,paperheight=7cm,top=1.5cm, bottom=1.5cm, left=1cm,right=1cm]{geometry}
5 %注意页面高度太小的话,可能会使[openright]失效,原因待研
6 %这里设置paperheight=6或者5都会使第一章开启页不在奇数页
7 \usepackage{xcolor}
8 \usepackage[pagestyles]{titlesec}
9 \usepackage{titletoc}
10 \titleformat{\chapter}{\zihao{4}\heiti}{\thechapter}{1em}{}
11 \titlespacing*{\chapter}{0pt}{0.0\baselineskip}{0.5\baselineskip}{0pt}
12 \titleformat{\section}{\zihao{5}\heiti}{\thesection}{1em}{}
13 \titlespacing*{\section}{0pt}{0.5\baselineskip}{0.5\baselineskip}{0pt}
14 \usepackage[backend=biber,style=gb7714-2015]{biblatex}
15 \addbibresource[location=local]{example.bib}
16 \defbibheading[subbibliography]{\bibname}{%
17 \section{#1}}%
18 \renewcommand{\bibfont}{\zihao{-6}\songti}
19 \setlength{\bibitemsep}{2pt}
20 \usepackage{hyperref}
21
22 \begin{document}
23 \tableofcontents

```

```

24
25 \begin{refsection}
26 \chapter{序章}
27 \titleformat{\section}[frame]{\normalfont}{\filright\footnotesize\enspace SECTION \thesection\enspace}
28 {8pt}{\bfseries\filcenter}
29 \section{序节}
30 序章内容\cite{GPS1988--}
31 \printbibliography[heading=subbibliography,title=本章参考文献]
32 \end{refsection}
33
34 \begin{refsection}
35 \chapter{正文一章}
36 \section{正文一节}
37 正文内容一\cite{杨洪升2013-56-75}
38 \printbibliography[heading=subbibliography,title=本章参考文献]
39 \end{refsection}
40 \end{document}

```

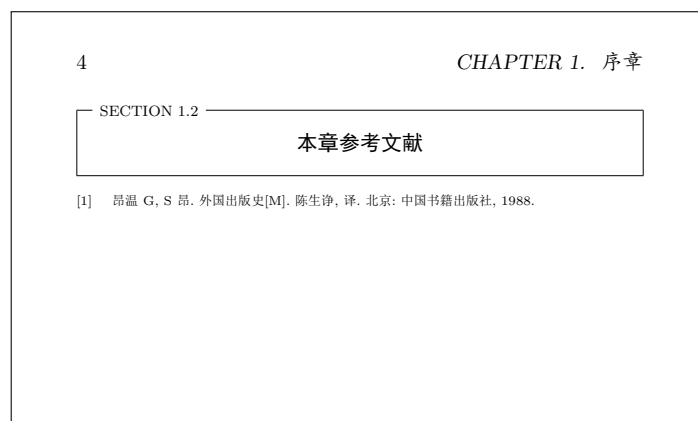


图 15 利用 titlesec 局部修改参考文献标题格式

## 2.8 参考文献表内容格式

参考文献内容的字体、字号、间距等格式可以重定义 biblatex 宏包提供的命令来实现。

### 2.8.1 一般设置方法

字体字号由钩子命令 bibfont 设置, 垂直间距由 bibitemsep、bibnamesep、bibinitsep 三个尺寸进行设置, bibitemsep 表示每一条参考文献之间的间隔, bibnamesep 表示当本条文献与前一条文献的责任者不同时设置的间隔, bibinitsep 表示本条文献与前一条文献的首字母不同时设置的间隔 (注意: 当设置排序 sorting=none 的时候, bibinitsep 的作用有所变化), 一般情况下使用 bibitemsep, bibnamesep 已然足够, 需要注意这三个尺寸遵守 addvspace 的规则, 同时存在时取最大的那个尺寸作为间隔。设置的方式如例37所示:

示例 37 参考文献表内容的格式

代码

```

1 %参考文献文本字体为默认字体, 字号为小五, 利用ctex设置
2 %如果不是利用ctex宏包, 可以利用其它字号设置命令
3 \renewcommand{\bibfont}{\zihao{6}}
4 %设置各条参考文献之间的间距为0pt
5 \setlength{\bibitemsep}{0pt}
6 %\setlength{\bibnamesep}{1ex}
7 %\setlength{\bibinitsep}{2ex}

```

如果还对文献表中的文献信息有更多字体、颜色、字号方面的格式设置需求,可以进一步利用域格式比如`\DeclareFieldFormat{booktitle}{\color{blue}\ttfamily#1}`进行设置, url 和 doi 字体也可以利用 `urlstyle`、`UrlFont` 等命令设置, 如例38所示。

示例 38 参考文献表内容的字体颜色格式

1 详见: `egfontinbiblio.tex`

代码

见`egfontinbiblio.tex`

另外, `biblatex-gb7714-2015` 样式包还增加了字体控制命令 `bibauthorfont`, `bibttitlefont`, `bibpubfont`, 用于控制文献表中作者、标题、出版项的字体和颜色。还注意在 `beamer` 类中, 因为 `beamer` 对 `biblatex` 做了 `patch`, 因此也可以使用 `beamer` 自身的方法进行字体和颜色控制, 当然使用这三个命令也没有问题。

## 2.8.2 段落格式和局部字体修改

参考文献表段落格式是由 `biblatex` 或者样式包提供的参考文献表环境打印和控制的, 通常是由 `list` 环境自定义而来。因此要设置参考文献表内容的段落格式, 需要从修改参考文献表环境 `defbibenvironment{bibliography}` 的重定义入手 (主要是修改 `list` 环境的参数), 当然通常情况这些是不需要去做的, 因为所选的样式包通常已经设置。

如果有特殊的需求也可以自定义一个参考文献表环境来控制其格式。如果仍然使用 `list` 通用环境作为基础环境来定义的话, 其段落格式的参数设置可以参考文献<sup>[15]74-80</sup><sup>[16]265-268</sup>。(注意到其实 `latex` 中的很多环境比如 `center`, `quote` 等都是由通用 `list` 环境所定义的。)

同时, 因为 `bibfont` 等命令使用具有局部性, 所以可以在文档中多次定义, 用于打印不同格式效果的文献表。例39给出一个完成的测试代码, 其中打印了 4 个参考文献表, 分别用了不同的字体。同时设置了 `bibitemsep`、`bibnamesep`、`bibinitsep` 三个尺寸控制各条文献的垂直间距。并且自定义了一个参考文献表环境 `marginref`, 并没有使用 `list` 环境, 而是简单的编组后打印环境, 可以看到其中各条文献的缩进效果。

`biblatex-gb7714-2015` 提供了更多的修改命令和接口, 可以实现更多的格式修改, 请参看: `biblatex-gb7714-2015.PDF`。

示例 39 参考文献表内容格式修改举例

1 `%file:egbibfont.tex`

代码

见`egbibfont.tex`

第一个文献表如图16所示, 使用的是默认字体即宋体。第二个文献表如图17所示, 使用字体是 `Arial` 和仿宋。第三个文献表如图18所示, 使用的是默认字体的等宽字体。第四个文献表如图19所示, 使用的是 `Courier New` 和楷书字体, 同时文献表环境是自定义的。(注意: 因为 `linux` 下没有 `Arial` 和 `Courier New` 字体, 所以用 `texlive` 自带的字体替换, `linux` 下使用 `texlive` 自带的字体需要特殊设置, 详见<sup>[11]15-16</sup>)。

## 2.8.3 文献中的特殊字符

一些参考文献中可能存在一些特殊的字符, 比如 `TEX` 相关的命令, 数学相关的字符等。这些字符通常用 `tex` 命令输入即可。然而有的时候参考文献的输出因为字体设置的原因导致特殊字符显示异常。比如`\textregistered{}`这个注册符号, 在不同的字体设置情况下, 正文中和文献表内的显示存在差异, 比如 `test-registered.tex`文件中注册符显示正常, 而更换字

参考文献

[1] 昂温 G, S 昂. 外国出版史[M]. 陈生诤, 译. 北京: 中国书籍出版社, 1988.

[2] 杨洪升. 四库馆私家抄校书考略[J]. 文献, 2013(1): 56-75.

[3] 马克思. 政治经济学批判[M]//马克思, 恩格斯. 马克思恩格斯全集: 第 35 卷. 北京: 人民出版社, 2013: 302.

[4] ANDERSEN J B, RAPPAPORT T S, YOSHIDA S. Propagation measurements and models for wireless communications channels[J]. IEEE Commun. Mag., 1995, 33(1): 42-49.

[5] BUSECK P R, NORD G L Jr., VEBLEN D R. Subsolidus phenomena in pyroxenes[M]//PREWITT C. Pyroxense. Washington, D.C.: Mineralogical Society of America, 1980: 117-211.

[6] CALKIN D, AGER A, THOMPSON M. A comparative risk assessment framework for wildland fire management: the 2010 cohesive strategy Science report:RMRS-GTR-262[R]. 2011: 8-9.

图 16 参考文献表格默认字体字号六

参考文献

[1] 昂温 G, S 昂. 外国出版史[M]. 陈生诤, 译. 北京: 中国书籍出版社, 1988.

[2] 杨洪升. 四库馆私家抄校书考略[J]. 文献, 2013(1): 56-75.

[3] 马克思. 政治经济学批判[M]//马克思, 恩格斯. 马克思恩格斯全集: 第 35 卷. 北京: 人民出版社, 2013: 302.

[4] ANDERSEN J B, RAPPAPORT T S, YOSHIDA S. Propagation measurements and models for wireless communications channels[J]. IEEE Commun. Mag., 1995, 33(1): 42-49.

图 17 参考文献表格仿宋字体字号六

参考文献

[1] 昂温 G, S 昂. 外国出版史[M]. 陈生诤, 译. 北京: 中国书籍出版社, 1988.

[2] 杨洪升. 四库馆私家抄校书考略[J]. 文献, 2013(1): 56-75.

[3] 马克思. 政治经济学批判[M]//马克思, 恩格斯. 马克思恩格斯全集: 第 35 卷. 北京: 人民出版社, 2013: 302.

[4] ANDERSEN J B, RAPPAPORT T S, YOSHIDA S. Propagation measurements and models for wireless communications channels[J]. IEEE Commun. Mag., 1995, 33(1): 42-49.

[5] BUSECK P R, NORD G L Jr., VEBLEN D R. Subsolidus phenomena in pyroxenes[M]//PREWITT C. Pyroxense.

图 18 参考文献表格等宽字体字号六

参考文献

昂温 G, S 昂. 外国出版史[M]. 陈生诤, 译. 北京: 中国书籍出版社, 1988.

杨洪升. 四库馆私家抄校书考略[J]. 文献, 2013(1): 56-75.

马克思. 政治经济学批判[M]//马克思, 恩格斯. 马克思恩格斯全集: 第 35 卷. 北京: 人民出版社, 2013: 302.

ANDERSEN J B, RAPPAPORT T S, YOSHIDA S. Propagation measurements and models for wireless communications channels[J]. IEEE Commun. Mag., 1995, 33(1): 42-49.

图 19 参考文献表格楷书字体字号六

体的文件 `test-registered-diff.tex` 中参考文献内的注册符与正文中的不一致。这是由于 biber 在处理文献信息的时候, 默认情况下, 首先会将一些设定的 tex 命令转换为 utf-8 字符便于比较和排序 (常见的转换见 `recode_data.xml` 文件), 因此在输出的时候如果设定的字体没有对应的 utf-8 字符, 那么就会显示异常。这个问题本质上是字体的问题, 但如果字体不能更换, 就需要利用 biber 来解决。biber 可以使用选项对输入和输出控制, 如果输入时 biber 不做转换, 或者输出时 biber 把 utf-8 字符转换回 tex 命令, 那么都能解决该问题。其中输入选项 `--decodecharsset=null` 表示需要转换的字符集为空, 即不做转换。而输出选项 `--output-safechars` 则能保证 biber 输出时将特殊字符转换回 tex 命令。更多细节详见 biber

文档。

特殊字符除了用 `tex` 命令输入外,有时也希望直接输入特殊字符本身,比如想在文献中直接输入希腊字母  $\alpha\beta$ ,而不是用数学环境中的命令`\alpha\beta`表示。但最后输出的时候并没有正确的输出。这个问题本质上仍然是字体设置问题,一般情况下文献表中的内容也采用与文档正文一致的字体,因此要直接显示类似希腊字母这样的特殊字符,使用 `xelatex` 编译时,就是需要有合适的字体支持,比如 CMU Serif 字体就支持 unicode 字符集。此时,只要设置全文档的英文主字体为 CMU Serif,那么就可以实现文献表总特殊字符的显示。

但有时正文文档使用的英文字体是不支持特殊字符的,那么可以在局部环境中使用支持特殊字符的字体,这利用前一节介绍过的 `bibfont` 的字体设置来实现。

#### 示例 40 文献中的特殊字符直接显示方法

代码

```
1 % 方法一：设置文档英文主字体，比如：
2 \usepackage{fontspec}
3 \setmainfont{CMU Serif}
4
5 % 方法二：局部利用其它支持的字体，比如：
6 \newcommand{\ftcmu}{\fontspec{CMU Serif}\selectfont}
7 \renewcommand{\bibfont}{\ftcmu}%设置英文字体不影响中文字体
8 \printbibliography
```

见 [egfontgreek.tex](#)

关于查找支持 unicode 字符的字体可以利用 `fc-list` 命令列出所有系统中的字体,然后查找字符串 `unicode`,一般带有该字符串的字体都是支持的,当然也可以一个一个字体的试,或者用字体查看软件来看字体文件的内容来确定。

#### 2.8.4 自定义环境举例

如第 2.8.2 节所述,可以利用 `defbibenvironment` 命令来自定义参考文献环境,其代码格式为如例 41 所示。其中还给出了顺序制样式的基本环境。对 `list` 的水平参数进行设置,以达到参考文献内容缩进对齐的效果。其中默认情况下,标签是右对齐的。如果要修改对齐方式为左对齐,可以有两种方式,一是设置域格式 `labelnumberwidth`,二是进一步重设 `makelabel` 命令,如例 42 所示,两种方式选其一即可。

然而我们看 GB/T 7714 2015 标准可以发现,其中顺序制的参考文献表中标签样式不是全部文献内容对齐,而是以当前参考文献项自身为基准进行对齐,这种方式如果通过 `list` 环境来实现修改起来会比较麻烦,因为其 `leftmargin` 需要根据每一项重设,熟悉 `list` 环境源码的朋友可以尝试一下。当然如果不使用 `list` 环境,我们可以利用 `hangindent` 或 `parshape` 来实现段落的格式。例 43 给出了本文作者的一种实现方式,其中需要重定义 `\blx@bibitem` 命令。详细的内容可以参考: [tex 源文件](#),其结果如图 20 所示,注意其中参考文献条目的顺序编码序号是用一个新的计数器处理的,事实上 `biblatex` 有提供相应的信息,利用该信息的一种实现可以参考 `biblatex-gb7714-2015` 样式包的 `gb7714-2015.bbx` 文件。

#### 示例 41 自定义参考文献表环境举例

代码

```
1 %defbibenvironment命令格式为:
2 \defbibenvironment{环境名}{环境开始代码}{环境结束代码}{各项的代码}
3
4 %使用list环境的defbibenvironment命令格式样例:
5 \defbibenvironment{bibliography}
6 {\list
```



---

```

7 {\printfield[labelnumberwidth]{labelnumber}}%标签, 由域labelnumber信息的提供
8 {\setlength{\labelwidth}{\labelnumberwidth}%标签宽度设置为labelnumberwidth
9 \setlength{\leftmargin}{\labelwidth}%
10 \setlength{\labelsep}{\biblabelsep}%标签与内容间距设置为\biblabelsep
11 \addtolength{\leftmargin}{\labelsep}%缩进宽度设置为\labelwidth+\labelsep
12 \setlength{\itemsep}{\bibitemsep}%垂直间距
13 \setlength{\parsep}{\bibparsep}}%
14 \renewcommand*{\makelabel}[1]{\hss##1}}
15 {\endlist}
16 {\item}

```

---

#### 示例 42 基于 list 的参考文献表标签的左对齐

代码

---

```

1 %方式一
2 \DeclareFieldFormat{labelnumberwidth}{\mkbibbrackets{#1}\hfill}
3 %方式二
4 \renewcommand*{\makelabel}[1]{\hss##1\hfill}

```

---

#### 示例 43 以当前条目为基准对齐的参考文献表环境

代码

---

```

1 %file:egbibenv.tex
2 \makeatletter
3 \def\blx@bibitem#1{%
4 \blx@ifdata{#1}
5 {\begingroup
6 \blx@getdata{#1}%
7 \blx@bibcheck
8 \iftoggle{blx@skipentry}{}%
9 \blx@setdefaultrefcontext{#1}%
10 \global\let\blx@noitem\@empty
11 \blx@setoptions@type\abx@field@entrytype
12 \blx@setoptions@entry
13 \blx@thelabelnumber
14 \addtocounter{instcount}{@ne
15 \blx@initsep
16 \blx@namesep
17 \csuse{blx@item@\blx@theenv}\relax
18 % \blx@initsep %移动到上面去, 恢复bibnamesep等机制的作用
19 % \blx@namesep
20 \csuse{blx@hook@bibitem}%
21 \blx@execute
22 \blx@initunit
23 \blx@anchor
24 \blx@beglangbib
25 \blx@sentence
26 \blx@pagetracker
27 \blx@driver\abx@field@entrytype
28 \blx@postpunct
29 \blx@endlangbib}%
30 \par\endgroup}%这里增加了一个\par
31 {}
32 \makeatother
33 \newcommand{\itemcmd}{%
34 \stepcounter{bibentrynumber}
35 \settowidth{\lengthid}{\arabic{bibentrynumber}}
36 \addtolength{\lengthid}{\biblabelsep}
37 \setlength{\lengthlw}{\textwidth}
38 \addtolength{\lengthlw}{-\lengthid}

```

```

39 \addvspace{\bibitemsep}%恢复\bibitemsep的作用
40 %\parshape 2 0em \textwidth \lengthid \lengthlw
41 \hangindent\lengthid
42 [\arabic{bibentrynumber}]\hspace{\biblabelsep}}
43 \newcounter{bibentrynumber}
44 \newlength{\lengthid}
45 \newlength{\lengthlw}
46 \defbibenvironment{envtest}
47 {\begingroup\setlength{\parindent}{0em}\setcounter{bibentrynumber}{0}}
48 {\endgroup}
49 {\itemcmd}%\newline\itemcmd

```

- |   |
|---|
| <p>[4] ANDERSEN J B, RAPPAPORT T S, YOSHIDA S. Propagation measurements and models for wireless communications channels[J]. IEEE Commun. Mag., 1995, 33(1): 42-49.</p> <p>[5] BUSECK P R, NORD G L Jr., VEBLEN D R. Subsolidus phenomena in pyroxenes[M]//PREWITT C. Pyroxense. Washington, D.C.: Mineralogical Society of America, 1980: 117-211.</p> <p>[6] CALKIN D, AGER A, THOMPSON M. A comparative risk assessment framework for wildland fire management: the 2010 cohesive strategy Science report:RMRS-GTR-262[R]. 2011: 8-9.</p> <p>[7] Parsons. The Mobile Radio Propagation Channel[M]. 2nd ed. Hoboken: John Wiley, Sons, 2000.</p> <p>[8] PARSONS J D. The Mobile Radio Propagation Channel[M]. 2nd ed. Hoboken: John Wiley, Sons, 2000.</p> <p>[9] PARSONS J D. The Mobile Radio Propagation Channel[M]. 2nd ed. [S.l.]: John Wiley, Sons, 2000.</p> <p>[10] PARSONS J D. The Mobile Radio Propagation Channel[M]. 2nd ed. Hoboken: John Wiley, Sons.</p> <p>[11] 宋史卷三: 本纪第三[M]//宋史: 第 1 册. 北京: 中华书局, 1977: 49.</p> <p>[12] 亚洲地质图编目组. 亚洲地层与地质历史概述[J]. 地质学报, 1978, 3: 194-208.</p> <p>[13] 陈晋镡, 张惠民, 朱士兴, 等. 蓟县震旦亚界研究[M]//中国地质科学院天津地质矿产研究所. 中国震旦亚界. 天津: 天津科学出版社, 1980: 56-114.</p> |
|---|

图 20 以当前条目为基准对齐的参考文献表环境

### 2.8.5 表格形式的参考文献表

一般数据中的参考文献表从内容上看是文本字符串，从格式上看是由一个一个段落构成的。通常 latex 中的参考文献表是一个有 list 环境构成的文本块，内部的每一项是由 list 环境设置参数确定格式的段落。所谓我们调整文献表的段落格式是：(1) 通过对 thebibliography 环境重定义（过程中设置 list 环境的参数），这是传统参考文献生成的方法。(2) 通过定义 defbibenvironment 命令重定义 bibliography 环境来实现的，这是基于 biblatex 生成参考文献的方法。

而我们有时有需求希望将参考文献放大一个框架里面，如果参考文献表作为整体放入，那么实现是比较简单的，只要在在参考文献表外面加上需要的框架即可。然而如果要将参考文献表中的每条参考文献单独的放入框架中，比如将参考文献表以表格形式来显示，即将每一条参考文献作为表格的一行来显示，显然这是一个比较麻烦的问题。

传统参考文献生成的方法，由于 thebibliography 环境是显示的核心，必须要对其做定义修改，比如在其内部使用表格，同时仍然兼具文献引用关联和超链接的功能。对于每条参考

文献则需要显式的给出表格内容所需格式即加入`&`、`\\`、`\hline`等命令,这通过手写或者修改 `bst` 样式来直接在 `bb1` 文件中输出。上述虽然应该都可以实现,但确是一个相当复杂的。

而基于 `biblatex` 生成参考文献的方法,由于其完全获取参考文献信息的特点在这种复杂任务中是有优势的,可以比较方便的实现上述目标)。下面我们来介绍怎么操作。

`biblatex` 生成参考文献完全是在 `tex` 环境中进行,因此每一条参考文献的输出实质上是一个递归的解析数据并输出数据的过程,在这个过程中如果我们能够在输出的过程中添加上表格格式的标识命令,并在全部参考文献信息外面套上一个表格环境就能够得到表格形式的参考文献表。而这个功能是由 `biblatex-ext` 宏包完成了,因此对于我们使用者来说,问题就会简单得多了,只要学会使用即可。

表格形式的文献输出命令为: `\printbibtabular`, 其参数类似于 `\printbibliography`, 用法也相同。

其输出格式本质上是由 `\defbibtabular` 定义的 `bibtabular` 环境决定的,所以我们要生成什么样的表格就是要定义什么样的 `bibtabular` 环境。

比如:

示例 44 一个简单的表格式参考文献表设置

代码

```
1 \defbibtabular{bibtabular}
2   {\begin{longtable}{|c|p{0.85\textwidth}}|}
3     \hline
4     序号 & 文献目录 \\
5     \hline
6   {\end{longtable}}
7   {\anchor{\printfield{labelnumber}} & \driver{}}\\ \hline}
```

基于上述设置,我们就可以利用 `printbibtabular` 输出一个两列的简单表格。

具体实践代码见: <https://blog.csdn.net/xenonhu/article/details/121696116>

示例参考: `biblatex-ext` 宏包提供实现了表格化的功能,我修改后的国防科大开题报告模板 (在 github 上)。

### 2.8.6 参考文献表行溢出问题

参考文献表中各条参考文献有可能长度大于文本宽度,因此会涉及到断行的问题。一般情况下,断行不会出现问题。但有的时候,特别是在 `biblatex-gb7714-2015 1.0g` 版以前,诸如 `inbook` 等类型标题标识符后的 `//` 后没有空格导致其后紧跟的单词无法断行,进而导致行的溢出。这种情况下,可以采用设置更大的 `tolerance` 值来解决,或者可以利用命令 `hyphenation` 指定断行点。这些设置可以与文献表打印命令放在一个编组内,如例45所示。

当因为 `url` 网址过长及断行不当导致溢出时,可以通过设置三个阈值来让其在数字,大写字母,小写字母后面断行,如例46所示。有时尽管 `url` 没有导致行溢出,但也设置 `url` 断行阈值,这也是为了让 `url` 内容更紧凑避免有些地方出现较大空格导致不美观。

当然 `biblatex` 还提供了一些阈值来增加断行的灵活性,比如 `abbrvpenalty`、`highnamepenalty`、`lownamepenalty` 等,但一般不需要使用。

示例 45 参考文献表行溢出问题解决

代码

```
1 {
2   %\hyphenation{Proce-edings}
3   \hyphenpenalty=5000 %断词阈值, 值越大越不容易出现断词}
```

```

4 \tolerance=500 %丑度, 10000为最大无溢出盒子, 参考the texbook 第6章
5 \hbadness=100 %如果丑度超过hbadness这一阈值, 那么就会发出警告
6 \printbibliography[heading=subbibintoc,title=【参考文献】]
7 }

```

示例 46 参考文献表 url 导致行溢出问题解决

代码

```

1 {
2 \setcounter{biburlnumpenalty}{100}
3 \setcounter{biburlucpenalty}{100}
4 \setcounter{biburllcpenalty}{100}
5 }

```

## 2.8.7 参考文献表内断行和分页问题

如果把文献表中的一个参考文献条目看做一个段落, 有时我们不希望在这个段落中分页, 而让该参考文献完整的处于一个页面中, 无论是在前一页底部还是后一页的顶部。这个段落分页的机制, 本质是有 tex 内部机制控制的, 类似于上一小节说的断行问题, 可以通过设置内部的惩罚阈值来控制。

主要是三个阈值: `\interlinepenalty` 用于控制在段落内分页的期望程度。 `\clubpenalty` 用于附加控制在段落的第一行分页的期望程度。 `\widowpenalty` 用于附加控制在段落的最后一行分页的期望程度。数值越小则越希望分开, 数值越大则越不希望分开, 设置为 10000, 在 tex 表示无限大, 就是强制不分开。

在 biblatex 中默认设置为:

示例 47 一个参考文献条目段落内部的断行

代码

```

1 \newcommand*{\bibsetup}{%
2 \interlinepenalty=5000\relax
3 \widowpenalty=10000\relax
4 \clubpenalty=10000\relax
5 \raggedbottom}

```

表示不鼓励在一个文献条目内断行, 但不禁止。如果一个条目小于 4 行, 则 tex 会总是让它处于一个页面内。如果超过 4 行, 那么会产生断行, 并保证在至少在 2 行之后断行。

这几个阈值最好与 `\raggedbottom` 一起使用。如果不想使用 `\raggedbottom`, 那么需要确保在 `\bibitemsep` 即条目间距里设置一些弹性尺寸, 以便实现良好的垂直布局。

## 2.8.8 英文文献中字符串的字母大小写问题

修改参考文献的各域的格式通常使用域格式进行控制, 因此英文文献条目各著录项的字母大小写转换也是类似。英文习惯的大小写方式主要有 5 种, 包括 Sentence case 句子模式 (一个句子中除首字母大写外其它全部小写); Title case 标题模式 (一个句子各单词首字母均大写); All caps 全大写 (一个句子全部大写); All lowercase 全小写 (一个句子全部小写); Small caps 小大写字母模式 (一个句子中除首字母大写外其它全部用 smallcap 字母)。

目前这五种方式都有可用的实现, 主要来自于 latex 核心代码, biblatex 宏包, mfirstuc 宏包。为了方便使用, 笔者做了梳理, 并采用规范的接口命令。包括:

`\MakeCapital{a small book of {SUN} rhyme}` 实现一个句子中首字母大写, 其字母不变的 Sentence case 模式;

`\MakeSentenceCase{a small book of {SUN} rhyme}` 实现一个句子中除首字母大写外其它全部小写的 Sentence case 模式;

`\MakeTitlecase{a small book of {SUN} rhyme}`实现一个句子各单词首字母均大写的 Title case 模式;

`\MakeUppercase{a small book of {SUN} rhyme}`实现一个句子全部大写的 All caps 模式;

`\MakeLowercase{a small book of {SUN} rhyme}`实现一个句子全部小写的 All lower-case 模式;

`\MakeSmallcaps{a small book of {SUN} rhyme}` 实现一个句子中除首字母大写外其它全部用 smallcap 字母的 Small caps 模式。

具体实现方式详见[latex 中英文字母大小写变换再讨论](#)。

需要注意的是在域格式使用中字母大小写转换的命令, 应在直接针对字符串的域格式中设置, 具体问题见: [Make the first letter of each word uppercase in sentence](#)。

### 2.8.9 英文文献中期刊名的缩写问题

有时某些出版物可能会要求在参考文献中使用期刊名的缩写。最直接的方式是在 bib 文件中输入期刊的缩写名, 但有时不希望去改 bib 文件, 只是在生成文献表的时候形成缩写。这时我们通常不采用类似 `jabref` 直接修改 bib 文件的方法。而采用动态修改的方式, 主要有:

1. 利用 `biblatex` 的动态数据修改, 在 `biblatex` 处理时将期刊全名替换为缩写名。
2. 增加新的域存放缩写名, 然后再 `biblatex` 的域输出时, 利用新的域替换需要缩写的域。

具体的实现见笔者之前写的文章: [biblatex 中参考文献期刊名缩写的实现](#)。

还有一种需求是在原有期刊(会议)名的后面添加缩写, 而不是替换, 那么也可以上述两种方式进行修改。对于第二种方式可以参考下面的示例48。实际上对于第二种方式也可以不增加新的域, 而是通过数据注解的方式添加信息, 至于输出的格式仍然需要修改, 参考<https://blog.csdn.net/xenonhu/article/details/120854271>。

示例 48 利用 `shortbooktitle` 域记录缩写通过样式调整添加

代码

```

1 \begin{filecontents*}{test.bib}
2 @Inproceedings{Nemec1997-209-214,
3   Title = {Force control of redundant robots},
4   Author = {B Nemec},
5   Booktitle = {Processings of Symposium on Robot Control},
6   shortbooktitle = {(PSRC)},
7   Pages = {209-214},
8   Country = {Nantes France},
9   Year = {1997}
10 }
11
12 @Article{Chiani1998-2998-3008,
13   Title = {Error probability for block codes over channels with block interference},
14   Author = {Chiani, M.},
15   Journal = {IEEE Trans. Inf. Theory},
16   shortbooktitle = {(ITIT)},
17   Number = {7},
18   Pages = {2998-3008},
19   Volume = {44},
20   Year = {1998}
21 }
22 \end{filecontents*}
```

---

```

23 \documentclass{ctexart}
24 \usepackage[backend=biber,style=gbt7714-2015]{biblatex}
25
26 \renewbibmacro*{maintitle+booktitle}{%
27   \usebibmacro{booktitle}\setunit{\addspace}\printfield{shortbooktitle}%
28   \newunit}
29
30 \renewbibmacro*{journal+issuetitle}{\bibpubfont%来自standard.bbx
31   \usebibmacro{journal}\setunit{\addspace}\printfield{shortbooktitle}%
32   %\setunit*{\addspace}%
33   \setunit*{\addcomma\addspace}%修改为增加一个逗号
34   \iffielddundef{series}
35     {}
36     {\newunit
37       \printfield{series}%
38       \setunit{\addspace}}%
39   %\usebibmacro{volume+number+eid}%
40   %\setunit{\addspace}%
41   \usebibmacro{issue+date}%
42   %\setunit{\addcolon\space}%
43   \iffielddundef{volume}{\setunit{\addcomma\space}}%
44   %换成逗号和空格
45   \usebibmacro{issue}%
46   \usebibmacro{volume+number+eid}%把卷期放到年份后面
47   %\newunit
48 }
49
50 \addbibresource[location=local]{test.bib}
51 \begin{document}
52   \nocite{*}
53   \printbibliography[heading=bibliography,title=参考文献]
54 \end{document}

```

---

讨论见: <https://github.com/CTeX-org/gbt7714-bibtex-style/issues/87>

#### 2.8.10 修改指定参数文献的格式

详见: <https://blog.csdn.net/xenonhu/article/details/120097980>

之前我写到博客上了, 有时间我把这些知识都整理进来。

#### 2.8.11 修改指定作者的格式

详见: <https://blog.csdn.net/xenonhu/article/details/120853247>

#### 2.8.12 修改文献表中日期的格式

详见: <https://blog.csdn.net/xenonhu/article/details/86933525>

#### 2.8.13 修改文献中标点的全半角形式 (中英文形式)

详见: <https://blog.csdn.net/xenonhu/article/details/124686522>

### 2.9 专题: 参考文献格式的修改

由于 biblatex 提供了参考文献数据的完全访问能力, 所以定义任何需要的样式都是可能的。因此用户一定可以选择合适的样式或者定制合适的样式来满足自身需要。当前对参考文献格式有一些特殊的需求, 且现成的类似的样式还无法满足要求时, 往往就需要进行定制, 比如 [17] 提出的问题。

在 latex 传统方法中, 当文献量较小时, 这种临时定制采用 thebibliography 环境是比较方便的, 因为 thebibliography 环境显式的将其中的内容插入, 因此需要什么样的格式, 那么



就在 `bibitem` 中输入指定格式的条目内容即可。但该环境存在一个问题即: 无论引用与否环境中的参考文献都会全部打印。该环境也可以用来生成多个参考文献表, 当然文献量较大时采用这种手动输入方式毕竟是繁琐的。我们使用 `biblatex` 就是为了避免这种繁琐的操作。

因为 `biblatex` 标准样式和大量第三方样式的存在, 定制往往是在这些样式基础上做的, 一般不需要从零开始设计样式, 而只要做一定修改和调整。`biblatex` 提供了两种比较快速的方式, 一是基于宏包选项的修改, 二是基于一些简单内容的格式修改。如果这两种方式还不能满足要求, 那么再考虑重新设计样式的问题。更复杂的样式设计问题详见第节。

### 2.9.1 通过宏包选项改变样式

在给定样式基础上, 可以使用宏包选项来定制一些需要的样式, 比如作者输出数量、文献表排序、文献标注排序、文献数据项的输出等等, 宏包选项的详细说明可以参见 [biblatex 手册中文版](#)。

**2.9.1.1 作者输出数量** 文献表和标注中的作者的数量可以通过宏包选项 `maxnames` 和 `minnames` 控制。两者也可以分开进行控制, 分别利用 `maxbibnames`, `minbibnames` 和 `maxcitenames`, `mincitenames` 进行控制。

**2.9.1.2 文献表排序** 文献表排序可以使用 `sorting` 选项进行排序, 有些样式中已经给出了 `sorting` 选项, 如果用户想要另外的排序方式, 那么可以自己选择合适 `sorting` 选项在 `biblatex` 宏包加载时给出。

**2.9.1.3 标注排序** 当一个引用命令中有多个参考文献引用关键字时, 可以利用 `sortcites` 选项来控制是否自动排序。当 `sortcites=false` 时不进行排序, 按给出的顺序排列。当 `sortcites=true` 时, 则按文献表的顺序排列。如果希望标注排序与文献表的排序不同, 可以利用新文境 `newrefcontext` 来实现。比如: 希望文献表按 `nyt` 排, 而标注用 `ynt` 排, 可以采用如下方式: 宏包加载时使用 `\usepackage[backend=biber,style=gb7714-2015ay,sortcite,sorting=ynt]{biblatex}`, 然后在文献打印前加 `\newrefcontext[sorting=nyt]`

**2.9.1.4 文献数据项的输出控制** 如果要取消 `url`, `doi`, `isbn` 等信息项的输出, 可以设置相应的选项为 `false`。比如: `url=false`。数据项的输出控制还可以从其它四个层次进行控制:

一是数据动态修改层次, 在动态数据修改中将数据去除。

#### 示例 49 利用动态数据修改来去除数据域

代码

```
1 \DeclareStyleSourcemap{
2   \maps[datatype=bibtex]{
3     \map{
4       \step[fieldset=url, null]
5     }
6   }
7 }
```

二是数据准备层次, 在数据应用前利用钩子将数据域去掉

#### 示例 50 利用钩子在数据应用前去除数据域

代码

```
1 \AtEveryBibitem{\clearfield{url}\clearlist{location}}
```

三是在驱动层次，直接修改 `dirver` 去掉数据域，或者利用选项来控制，前面利用选项的控制其实就是在这一层次。

四是在域输出宏层次，直接修改域的输出宏使其不输出即可。

三四两种涉及驱动和输出宏的修改，后面会有通用的介绍，这里不再说明。

**2.9.1.5 标注后面的标点控制** 在英文有时因为习惯问题会有标点的问题，比如句尾出现脚注的标签，需要把它放到句点后面。有时可能也需要把标注标签放到句点后面。这时需要用 `autopunct` 选项来扫描标注后面的标点。下面的例子给出了在句尾的脚注文献与标点的处理。

#### 示例 51 `autopunct` 选项与脚注之后句点处理

1 见文件：`testautopunct.tex`

代码

### 2.9.2 样式的临时定制

`biblatex` 可以实现简单快速的格式修改。`biblatex` 使用手册的 `Bibliography Style Files` 一节给出了样式文件定义的一般模式和方法。依据该模式是容易实现样式快速定制的，因为在一般情况下定制/修改样式文件，其实并不需要处理太多的内容。

总结一下，简单的样式修改通常会包括：

- (1) 加载定制所需的基础样式比如标准样式或其它样式；
- (2) 设置宏包选项；
- (3) 设置单元或块的标点；
- (4) 设置域格式；
- (5) 设置驱动格式；

当然对于一些复杂的参考文献样式（比如 `gb7714-2015`）还需要处理更多的内容，包括：

- (1) 增加和应用需要的判断和功能函数；
- (2) 增加一些域格式，增加修改应用本地化字符等；
- (3) 动态数据处理；
- (4) 增加宏包选项；
- (5) 等等。

而标注样式的定制则更为简单，主要处理内容包括：

- (1) 加载定制所需的基础样式比如标准样式或其它样式；
- (2) 设置宏包选项；
- (3) 定义或重定义引用命令；

这里以前述提到的 `ctex` 论坛上的提问为例进行讨论。该问题中参考文献著录格式主要需要修改的内容是分块、分块的标点、分块内容的格式。其中：作者是一个块作为一行，标题是一个块作为一行，其它是一个块作为一行，那么把块标点设置为换行，就可以实现多行。同时各块的内容的字体不同，作者是粗体，标题是等宽，其它是斜体，那么把对应的块所构成域的格式改成需要的格式即可。这里给出该问题的一个简单解答，如图 21 所示。采用的样式文件代码如例 52 所示，其中为了简化修改，把一些条目设置成了相同的格式。

#### 示例 52 参考文献样式定制举例

1 `\ProvidesFile{studf.bbx}[2016/12/07 v1.0e biblatex bibliography style]`  
 2 略，详见源文件。

代码

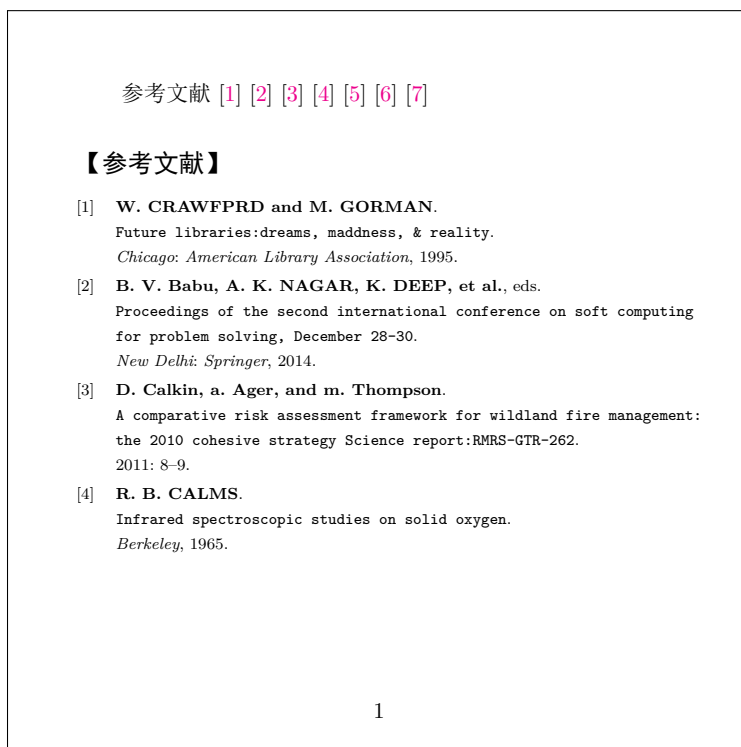


图 21 参考文献样式快速定制

源代码见[studf.bbx](#), 测试文件见[egstylecustomize.tex](#)。

如果对于中文参考文献需要一些特殊的处置, 那么可以基于 gb7714-2015 进行修改, 如例53给出的代码。其结果如图22所示。

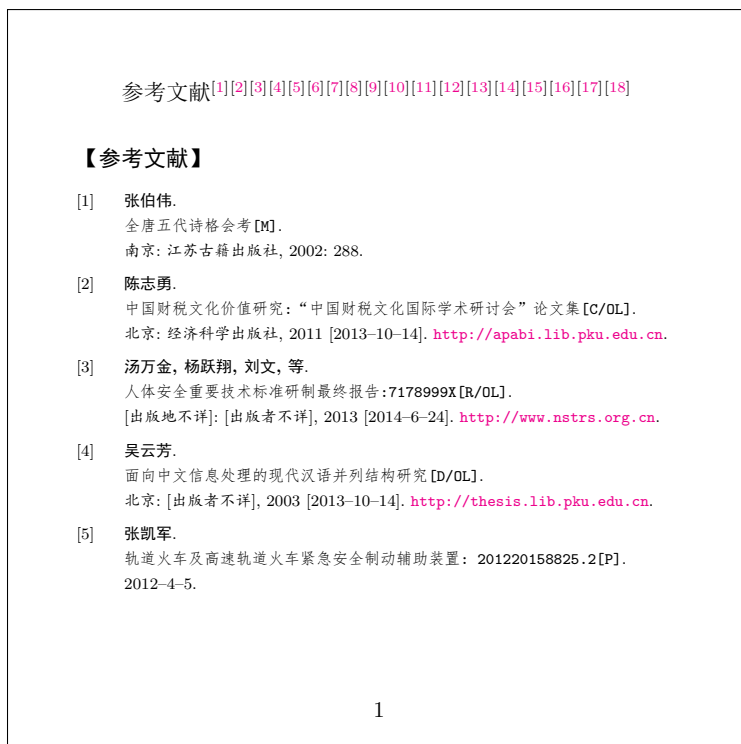


图 22 国标风格的参考文献样式快速定制 A

示例 53 参考文献样式定制举例-国标样式

---

```

1 \ProvidesFile{gbudf.bbx}[2016/12/07 v1.0e biblatex bibliography style]
2 略，详见源文件。

```

---

源代码见 [gbudf.bbx](#)，测试文件见 [egstylecustomizegb.tex](#)。

样式修改或定制中，是针对不同的内容，比如标点、标签、域格式、域顺序等等。不同的内容需要采用不同的方法，下面介绍一些常见内容的处理方法：

### 2.9.3 分隔符/定界符环境

分隔符/定界符其实就是常说的标点。分隔符 (定界符) 通常是全局定义的。也就是说，无论在哪里使用，打印的分隔符都是相同的。然而，如果想要在不同环境下中打印不同的符号，这些全局的分隔符就不可取了。biblatex 在利用 newcommand 定义分隔符的常规方式基础上，还提供了更为智能的分隔符环境 (context，可以是“标注文本内部”或者“参考文献项”等环境)，用以定义局部使用的分隔符。

biblatex 使用 DeclareDelimFormat 命令定义不同分隔符环境中的分隔符，利用 printdelim 环境在不同的环境中打印不同的分隔符。biblatex 在不同位置自动创建了若干个默认的 context：包括：(1) none 文档开始处。(2) bib 以 printbibliography 开始的参考文献内部或者 usedriver 内部。(3) biblist 以 printbiblist 开始的参考文献列表内部。(4) ‘citecommand’ 使用 DeclareCiteCommand 定义的 citecommand 引用命令内部。也可以使用 delimcontext 命令手动设置分隔符环境。

### 2.9.4 标签的格式和装饰

标签的格式控制主要是通过域格式实现，比如：

示例 54 标签的格式控制

---

```

1 \DeclareFieldFormat{labelnumberwidth}{\mkbibparens{#1}}
2 \DeclareFieldFormat{labelformat}{\mkbibbrackets{#1}}
3 \DeclareFieldFormat{labelnumber}{\mkbibitalic{#1}}

```

---

代码

此外，还有 labelalpha，extraalpha 等其它域的格式可以调整，更多的可以参考 biblatex 宏包的源代码。

标签的包围符号，一般是 []，但也可以其它的，有时可以全局的做修改，比如：

示例 55 标签的包围符号全局修改

---

```

1 \renewcommand*\bibleftbracket{()}
2 \renewcommand*\bibrightbracket{()\midsentence}

```

---

代码

也可以用另一种方式做局部修改，比如在引用的地方修改，比如：

示例 56 标签的包围符号局部修改

---

```

1 %利用()
2 \newrobustcmd*{\parentexttrack}[1]{%
3   \begingroup
4   \blx@blxinit
5   \blx@setsfcodes
6   \blx@bibopenparen#1\blx@bibcloseparen%
7   \endgroup}
8
9 \AtEveryCite{%
10  \let\parentext=\parentexttrack%

```

---

代码

```

11 \let\bibopenparen=\bibopenbracket
12 \let\bibcloseparen=\bibclosebracket}
13
14 %利用[]
15 \newrobustcmd*{\parentexttrack}[1]{%
16   \begingroup
17   \blx@blxinit
18   \blx@setsfcodes
19   \blx@bibopenbracket#1\blx@bibclosebracket%
20   \endgroup}
21
22 \AtEveryCite{%
23   \let\parentext=\parentexttrack%
24   \let\bibopenbracket=\bibopenparen%
25   \let\bibclosebracket=\bibcloseparen}

```

### 2.9.5 label 类的域及其应用

无论是在文献著录表中还是在引用标注中，常存在一个称为标签的东西。比如顺序编码制的文献表中每一个条目前的数字标签，如 [1]；又比如作者年制的引用标注标签中作者名和年份，如 (author 2001)。这些都与 biblatex 中一种类型的域相关，就是 label 类型的域。

首先我们看两个 label 类的域：shorthand 和 label。shorthand 域在标注样式中，用于替代常规标签的特殊标签。如果有定义，那么它会覆盖默认的标签。而 label 域在标注样式中，如果生成常规标签的所需数据均缺失，那么该域的内容可用来代替常规标签。例如，当作者年制标注样式要生成某个条目的标签，但该条目的作者或年份缺失，那么会使用后备的 label。shorthand 和 label 正好相反，label 只是作为后备而使用。基于这一认知，如果我们需要生成指定内容的标注标签，那么可以在 bib 文件中使用这两个域，当然这是作者年制的情形，对于顺序编码制生成的数字标签，并不会应用这两个域。

在顺序字母 (alpha) 制的标注样式中，标签默认由 labelname 列表抽取的首字母加上出版年的最后两个数字构成。而 label 域可用来重设它的非数字部分 (non-numericportion)。如果定义了 label 域，biblatex 将使用它的值加上出版年的后两个数字生成 labelalpha。而 shorthand 域也可用于重设整个标签。如果定义了该域，labelalpha 就是 shorthand 域，而不是一个自动生成的标签。

当然上述逻辑并不是绝对的，这只是默认的样式设定的逻辑，对于顺序字母 (alpha) 制的标注标签，用户也可以通过 \DeclareLabelalphaTemplate 命令修改前述逻辑。而作者年制等 \DeclareLabelname、\DeclareLabeldate、\DeclareExtradata、\DeclareLabeltitle 等进行调整。

shortauthor、shorteditor、shortjournal、shortseries、shorttitle 等 label 类的域，也有一些类似的性质。当给出 shortauthor、shorteditor 时，默认的标注样式会在所有标注里使用短名称，比如：shortauthor 而在参考文献表中打印全名，比如 author。

而顺序编码标签使用的 labelnumber 域，是由 biblatex 自动赋值的。

在文献表中，处在不同分节的文献可以在标签前加上前缀 (labelprefix) 来区分。

## 2.10 专题：筛选、缩略信息、排序、索引

### 2.10.1 文献分类筛选的不同方法

除了前述的分章节打印外参考文献可以分类筛选打印，主要基于 printbibliography 命令的选项来实现。主要包括：

- type
- notype
- keyword
- notkeyword
- category
- notcategory
- filter
- check

其中 category, filter, check 是需要手动定义的。

type, notype 分别表示打印是和不是某种条目类型的文献, keyword, notkeyword 分别表示打印 keywords 域中包含或不包含某一个关键词的文献, category, notcategory 分别表示是或不是某一设定的 category 的文献, filter 表示属于某一设定筛选器的文献。使用时注意, category 必须要在导言区定义。check 则是更底层的命令, 常用在自定义的文献表中。

### 2.10.2 筛选打印示例

例57给出了这些选项的测试, 分别利用 type 和 keyword 区分了图书等文献, 定义了 collections 和 standard 两个筛选器用于打印论文集会议录和标准文献, 定义了名为 reportandthesis 的 category 用于打印学位论文和报告。结果如图23所示。测试文档见: [egbibfilter.tex](#)

示例 57 分类筛选打印参考文献表

---

```

1 %file:egbibfilter
2 \DeclareBibliographyCategory{reportandthesis}
3 \addtocategory{reportandthesis}{汤万金2013-09-30--,Calkin2011-8-9,吴云芳2003--,CALMS1965--}
4
5 \printbibliography[type=book,notkeyword=standard,title=【普通图书】]
6
7 \defbibfilter{collections}{%
8   type=collection
9   or type=proceedings
10  or type=incollection
11  or type=inproceedings
12 }
13 \printbibliography[filter=collections,title=【论文集、会议录】]
14
15 \defbibfilter{standard}{%
16   ( type=book or type=inbook )
17   and keyword=standard
18 }
19 \printbibliography[filter=standard,title=【标准文献】]
20 \printbibliography[type=inbook,notkeyword=standard,title=【专著中析出的文献】]
21 \printbibliography[type=article,notkeyword=news,title=【期刊中析出的文献】]
22 \printbibliography[type=article,keyword=news,title=【报纸析出的文献】]
23 \printbibliography[category=reportandthesis,title=【报告和学位论文】]
24 \printbibliography[notcategory=reportandthesis,title=【非报告和学位论文】]

```

---

代码

例58则展示了 category 和 check 筛选, 见测试文件: [testfilterCategory.tex](#), [testfilterCheck.tex](#)。

示例 58 category 和 check 筛选

---

```

1 %category
2 \DeclareBibliographyCategory{kindsofnameformats}
3 \addtocategory{kindsofnameformats}{zhang2009china,xu2012role,wang2006another,

```

---

代码



文献<sup>[1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18]</sup>

#### 【普通图书】

- [1] 张伯伟. 全唐五代诗格会考[M]. 南京: 江苏古籍出版社, 2002: 288.
- [2] CRAWFPRD W, GORMAN M. Future libraries:dreams, madness, & reality[M]. Chicago: American Library Association, 1995.

#### 【论文集、会议录】

- [3] 陈志勇. 中国财税文化价值研究: “中国财税文化国际学术研讨会”论文集[C/OL]. 北京: 经济科学出版社, 2011 [2013-10-14]. <http://apabi.lib.pku.edu.cn>.
- [4] BABU B V, NAGAR A K, DEEP K, et al. Proceedings of the second international conference on soft computing for problem solving, December 28-30[C]. New Delhi: Springer, 2014.

#### 【标准文献】

- [11] 全国广播电视标准化技术委员会. 广播电视音像资料编目规范: 第 2 部分 广播资料-GY/T202.2-2007[S]. 北京: 国家广播电影电视局广播电视规划院, 2007: 1.
- [12] 国家环境保护局科技标准司. 土壤环境质量标准:GB 15616-1995[S/OL]. 北京: 中国标准出版社, 1996: 2-3 [2013-10-14]. <http://wenku.baidu.com>.

#### 【专著中析出的文献】

- [13] 楼梦麟, 杨燕. 汶川地震基岩地震动特征分析[M/OL]//同济大学土木工程防灾国家重点实验室. 汶川地震震害研究. 上海: 同济大学出版社, 2011: 11-12 [2013-05-09]. <http://apabi.lib.pku.edu.cn>.
- [14] BUSECK P R, NORD G L Jr., VELEN D R. Subsolidus phenomena in pyroxenes[M]//PREWITT C. Pyroxene. Washington, D.C.: Mineralogical Society of America, 1980: 117-211.

#### 【期刊中析出的文献】

- [15] 陈建军. 从数字地球到智慧地球[J/OL]. 国图资源导刊, 2010, 7(10): 93 [2013-03-20]. <http://d.g.wanfangdata.com.cn>. DOI: 10.3969/j.issn.1672-5603.2010.10.038.
- [16] DES MARAIS D J, STRAUSS H, SUMMONS R E, et al. Carbon isotope evidence for the stepwise oxidation of the Proterozoic environment[J]. Nature, 1992, 359: 605-609.

#### 【报告和学位论文】

- [5] 汤万金, 杨跃翔, 刘文, 等. 人体安全重要技术标准研制最终报告:7178999X[R/OL]. 2013 [2014-06-24]. <http://www.nstrs.org.cn>.
- [6] CALKIN D, AGER A, THOMPSON M. A comparative risk assessment framework for wildland fire management: the 2010 cohesive strategy Science report:RMRS-GTR-262[R]. 2011: 8-9.
- [7] 吴云芳. 面向中文信息处理的现代汉语并列结构研究[D/OL]. 北京: 北京大学, 2003 [2013-10-14]. <http://thesis.lib.pku.edu.cn>.
- [8] CALMS R B. Infrared spectroscopic studies on solid oxygen[D]. Berkeley: Univ. of California, 1965.

#### 【非报告和学位论文】

- [1] 张伯伟. 全唐五代诗格会考[M]. 南京: 江苏古籍出版社, 2002: 288.
- [2] CRAWFPRD W, GORMAN M. Future libraries:dreams, madness, & reality[M]. Chicago: American Library Association, 1995.
- [3] 陈志勇. 中国财税文化价值研究: “中国财税文化国际学术研讨会”论文集[C/OL]. 北京: 经济科学出版社, 2011 [2013-10-14]. <http://apabi.lib.pku.edu.cn>.
- [4] BABU B V, NAGAR A K, DEEP K, et al. Proceedings of the second international conference on soft computing for problem solving, December 28-30[C]. New Delhi: Springer, 2014.
- [9] 张凯军. 轨道交通火车紧急安全制动辅助装置: 201220158825.2[P]. 2012-04-05.
- [10] KOSEKI A, MOMOSE H, KAWAHITO M, et al. Compiler: US828402[P/OL]. 2002-05-25 [2002-05-28]. <http://FF&p=1&u=metahtml/PTO/search-bool.html&r=5&f=G&l=50&col=AND&d=PG01&sl=IBM.AS.&oS=AN/IBM&RS=AN/IBM>.
- [11] 全国广播电视标准化技术委员会. 广播电视音像资料编目规范: 第 2 部分 广播资料-GY/T202.2-2007[S]. 北京: 国家广播电影电视局广播电视规划院, 2007: 1.
- [12] 国家环境保护局科技标准司. 土壤环境质量标准:GB 15616-1995[S/OL]. 北京: 中国标准出版社, 1996: 2-3 [2013-10-14]. <http://wenku.baidu.com>.
- [13] 楼梦麟, 杨燕. 汶川地震基岩地震动特征分析[M/OL]//同济大学土木工程防灾国家重点实验室. 汶川地震震害研究. 上海: 同济大学出版社, 2011: 11-12 [2013-05-09]. <http://apabi.lib.pku.edu.cn>.
- [14] BUSECK P R, NORD G L Jr., VELEN D R. Subsolidus phenomena in pyroxenes[M]//PREWITT C. Pyroxene. Washington, D.C.: Mineralogical Society of America, 1980: 117-211.
- [15] 陈建军. 从数字地球到智慧地球[J/OL]. 国图资源导刊, 2010, 7(10): 93 [2013-03-20]. <http://d.g.wanfangdata.com.cn>. DOI: 10.3969/j.issn.1672-5603.2010.10.038.

1

图 23 文献表筛选打印

```

4 calvo2006phoenix,bernanke1989agency,adam2011house,fazzari1993working}
5
6 \printbibliography[category=kindsofnameformats,title=五种不同的姓名格式]
7 \printbibliography[notcategory=kindsofnameformats,title=某种期刊的姓名格式]
8
9 %check
10 \defbibcheck{uppercase}{%
11 \iffielddundef{nameformat}{}\skipentry}}
12
13 \defbibcheck{lowercase}{%
14 \iffieldequalstr{nameformat}{lowercase}{}\skipentry}}
15
16 \defbibcheck{givenahead}{%
17 \iffieldequalstr{nameformat}{givenahead}{}\skipentry}}
18
19 \defbibcheck{familyahead}{%
20 \iffieldequalstr{nameformat}{familyahead}{}\skipentry}}
21
22 \defbibcheck{pinyin}{%
23 \iffieldequalstr{nameformat}{pinyin}{}\skipentry}}

```

```

24
25 \printbibliography[check=uppercase,title=国标风格-全大写]
26 \printbibliography[check=lowercase,title=国标风格-大小写不变]
27 \printbibliography[check=pinyin,title=中文常见风格-拼音]
28 \printbibliography[check=givenahead,title=biblatex标准风格-名在前]
29 \printbibliography[check=familyahead,title=biblatex标准风格-姓在前]

```

### 2.10.3 筛选的多个文献表的顺序编码标签

biblatex 与标准的 LaTeX 不同，默认情况下，biblatex 宏包生成的顺序编码标签在文档正文开始处就分配给文献表的全体文献。因此当有多个筛选文献表时可能会出现文献表中的顺序编码不连续的情况，这可以利用宏包的 `defernumbers` 选项解决。

但如果 `defernumbers` 选项被激活，各文献的顺序编码标签会在任意文献表中被打印时才会做第一次分配。

该选项需要在后端将数据导出到 `bbl` 文件后再运行两次 LaTeX（除由分页变化等要求的编译外）。需注意的一个要点是，如果你在文档中改变了该选项值（或者那些依赖于本选项的选项值，例如与 `printbibliography` 宏相关的选项，），那么很可能需要删除当前的辅助文件然后重新运行 L<sup>A</sup>T<sub>E</sub>X 来获得正确的顺序编码。需要记住 `defernumbers` 选项只与顺序编码标签有关，而与文献表中文献的排序无关。主要用于解决做文献筛选后顺序编码不连续的问题。如例59所示：

示例 59 多个筛选文献表的顺序编码

1 略，详见 `testdefernumber.tex` 文档

代码

文档为：[testdefernumber.tex](#)

### 2.10.4 文献缩略信息打印

参考文献处理中有不同的缩略信息可以打印，比如整篇文献的缩略词或者说是代替常规标签的缩略标签 (`shorthand`)，集体作者的缩写形式 (`shortauthor/shorteditor`)，期刊名的缩写形式或其首字母缩略语 (`shortjournal`)，`series` 的缩写形式或其首字母缩略语 (`shortseries`)，标题的缩略形式 (`shorttitle`) 等。

这些信息可以通过 `printshorthands`，`printbiblist` 来输出。`printshorthands` 一般用于存在 `shorthands` 域的情况。而 `printbiblist` 用于自定义输出信息的情况，可以使用 `short` 相关域也可以不使用。

例60给出了输出 `booktitle` 域信息的示例。其中要使用 `printbiblist` 输出自定义信息，必须要定义对应的驱动器。而相应的过滤器，著录环境，排序格式，标题格式等可以定义或不定义。结果如图24所示。

示例 60 论文集信息打印

1 %见文件：`egtestshort.tex`

代码

例61给出了使用默认的 `short` 系列域的方式，主要输出 `title` 和 `journal` 域的信息。结果如图25所示。

示例 61 标题和期刊信息打印

1 %见文件：`egtestshorta.tex`

代码

编译信息:biblatex v3.12;gb7714-2015\*.\* version number 2018/12/22 v1.0o.

## 1 文献表

文献<sup>[1][2][3][4][5]</sup>

## 文集名

Proc. IEEE Symp. on New Frontiers in Dynamic Spectrum Access Networks  
Proc. IEEE Symp. on New Frontiers in Dynamic Spectrum Access Networks  
Symposium on Applications of Holography in Mechanics  
中国科协学术期刊编辑工作经验交流会资料选

## 文集名使用指定环境

DySPAN Proc. IEEE Symp. on New Frontiers in Dynamic Spectrum Access Networks  
DySPAN Proc. IEEE Symp. on New Frontiers in Dynamic Spectrum Access Networks  
AHM Symposium on Applications of Holography in Mechanics  
科协学术 中国科协学术期刊编辑工作经验交流会资料选

图 24 论文集信息打印

## Abbreviations

JSTH John Tumble (1946). "A Tremendously Interesting Idea". In: *Journal of Tremendously Interesting Ideas* 1.1.  
JSTIP John Smith (1947). "A Tremendously Interesting Opinion". In: *Journal of Tremendously Interesting Opinions* 1.1.

## Journal Shorthands

JTIO *Journal of Tremendously Interesting Opinions*  
JTII *Journal of Tremendously Interesting Ideas*

## Title Shorthands

TIP "A Tremendously Interesting Opinion"  
TII "A Tremendously Interesting Idea"

## References

Smith, John (1947). "A Tremendously Interesting Opinion". In: *Journal of Tremendously Interesting Opinions* 1.1.  
Tumble, John (1946). "A Tremendously Interesting Idea". In: *Journal of Tremendously Interesting Ideas* 1.1.

图 25 标题和期刊信息打印

例62给出了在一个表中同时输出 shorthand 和 shortseries 的方式, 主要输出 title 和 journal 域的信息。结果如图26所示。

示例 62 不同缩略信息在一个表中打印

1 %见文件: egtestshortb.tex

代码

### 2.10.5 在标注和文献表中使用缩略信息

上一小节, 讨论的时把缩略信息作为一个单独的部分来输出以强调这些内容。但有的时候可能需要在文献表或标注中使用缩略信息, 而这又是另一个概念了。

latex 中的参考文献信息由 bib 文件提供, 因此除非管理 bib 文件的软件有增加缩略信息域的功能, 否则就需要手动的在 bib 文件中输入, 比如有 title={world health organization}, 那么要使用缩略信息, 就要输入, shorttitle={WHO}。

如果我们要在文献表中使用缩略信息, 有两种主要的方法:

(Talbert 1992) (Hofius 1989) ( <i>TDNT</i> )
<b>Abbreviations</b>
<i>TDNT</i> Kittel, Gerhard and Gerhard Friedrich, eds. (1964–1976). <i>Theological Dictionary of the New Testament</i> . Trans. by Geoffrey W. Bromiley. 10 vols. Grand Rapids: Eerdmans
<i>WUNT</i> Wissenschaftliche Untersuchungen zum Neuen Testament
<b>References</b>
Hofius, Otfried (1989). <i>Paulusstudien</i> . Wissenschaftliche Untersuchungen zum Neuen Testament 51. Tübingen: Mohr Siebeck.
Kittel, Gerhard and Gerhard Friedrich, eds. (1964–1976). <i>Theological Dictionary of the New Testament</i> . Trans. by Geoffrey W. Bromiley. 10 vols. Grand Rapids: Eerdmans.
Talbert, Charles H. (1992). <i>Reading John: A Literary and Theological Commentary on the Fourth Gospel and the Johannine Epistles</i> . New York: Crossroad.

图 26 不同缩略信息在一个表中打印

一是重定义使用相应信息的宏,比如`\renewbibmacro*{journal}`,可以将其中的 `journaltitle` 域替换为 `shortjournal` 域。

二是使用域信息的存储和恢复,比如:

示例 63 域信息替换方法

```
1 \savefield{shortjournal}{\temptitle}%
2 \restorefield{journaltitle}{\temptitle}%
```

代码

如果要在不同的位置比如在标注中,或者在文献表中使用,那么利用 `biblatex` 的局部编组特性,可以利用钩子 `AtEveryCitekey` 等方式来实现。看下面一个具体的例子,结果如图27所示。

示例 64 在标注和著录表中使用缩略信息

```
1 %见文件: egtestshortc.tex
```

代码

Some text. <sup>1</sup>
<b>Abbreviations</b>
EP <i>Episepsis</i> . 1970–.
<b>References</b>
<b>Les sources principaux</b>
<i>Episepsis</i> . 1970–.
<b>Les sources secondaires</b>
“Communique de la Commission”. In: EP 36 (1971), pp. 8–9.
“La Commission préparatoire”. In: EP 36 (1971), p. 2.
“La première Conférence”. In: EP 54 (1972), pp. 2–5.
<sup>1</sup> “La Commission préparatoire”. In: EP 36 (1971).

图 27 不同缩略信息在一个表中打印

### 2.10.6 文献信息的动态/静态替换

上一节使用缩略信息的方法中, 我们首先要给出缩略信息, 使用和输入这一信息, 还有一些方便的工具和方法, 比如使用信息的动态/静态修改/替换。

信息 (除了域信息也包括条目信息) 的处理包括替换方法, 除了前一节介绍的 `savefield-/restorefield` 外还包括: `crossref`、`xref`、`xdata`、`@string`、`sourcemap` 等, 共 6 种方法。下面进行一一介绍:

`crossref` 域用于指定要继承的父对象, 当比如在一个 `inbook` 条目中给出一个 `crossref` 域指定一个 `book` 条目, 那么该 `inbook` 条目就会从该 `book` 条目继承 `bookauthor`, `booktitle` 信息。而 `xref` 域用于指定条目的父子关系, 当指定父对象的次数超过一定的阈值时, 父条目会被加入到文献表中, 这是一种条目处理方式。当然 `crossref` 域指定的父对象也有根据阈值加入文献表的特性。使用方法可以参考手册第 2.4.1 节, 继承的规则和机制也可以自定义, 参考手册第 4.5.11 节。

`xdata` 是一种条目类型, 作为数据容器用, 该类型定义的条目的可以被继承。利用 `xdata` 类型定义一个条目, 其中包含要被继承的信息, 然后在需要这些信息的条目中加入域 `xdata`, 指定需要的 `xdata` 条目的关键词, 就可以将信息继承过来。而且条目中的 `xdata` 域支持分隔值列表, 因此可以从多个条目继承信息。具体方法参考手册第 3.13.6 节。

`@string` 方法是与 `xdata` 域有点类似的方法。在 `bib` 文件中定义一个 `@string`, 其内容包括一个字符串标签和字符串内容。在其它文献条目中可以把域指定为该字符串标签, 用来继承该字符串内容。例 65 给出了一个例子。更多的内容详见 Paul Stanley 的教程 ([pdf 文档](#))。

示例 65 string 方法继承数据

代码

```
1 @string{NYULSQ="NYUniversityLegalStudiesQ"}
2 @Article{art1,
3   title = {article title},
4   journaltitle= NYULSQ,
5   date = {1971},
6   number = {36},
7   pages = {8--9},
8 }
```

最后一个是最重要, 最常用, 最强大的 `sourcemap`, 即 `biber` 的动态数据修改。`biblatex` 提供了 4 个层次的数据动态修改, 常用的是用户层、样式层、驱动层这三层的 `sourcemap`。`sourcemap` 主要包括如下功能及其组合:

- 将数据源条目类型映射为其它类型
- 将数据源域映射为其它域
- 给条目添加新域
- 从条目移除域
- 用标准的 Perl 正则表达式匹配和替换, 修改域的内容。
- 将上述操作限制在来自特定数据源的条目
- 将上述操作限制在某些条目类型。
- 将上述操作限制在某一特定的参考文献节。

根据这些功能描述, 就可以知道对于 `bib` 文件中域的信息处理, 对于 `sourcemap` 来说只是很自然的一件事情。例 66 给出了将期刊名用其缩略信息来代替的三种操作方式。其中第一种和第二种本质是一样的, 就是将 `journaltitle` 设置为 `shortjournal` 的内容, 只是写法不同。

第三种操作，则做了匹配和替换，当匹配特定的目标时，才做替换。关于 sourcemap 的更多内容详见 biblatex 手册，第 4.5.3 节。

#### 示例 66 sourcemap 处理域信息

代码

```

1 \DeclareSourcemap{
2   \maps[datatype=bibtex]{
3     \map[overwrite]{ % Notice the overwrite: replace one field with another
4       \step[fieldsource=shortjournal,fieldtarget=journaltitle]
5     }
6   }
7 }
8
9 \DeclareSourcemap{
10  \maps[datatype=bibtex]{
11    % Journal abbreviations
12    \map[overwrite]{
13      \step[fieldsource=shortjournal]
14      \step[fieldset=journaltitle,origfieldval]
15    }
16  }
17 }
18
19 \DeclareSourcemap{
20  \maps[datatype=bibtex]{
21    % Journal abbreviations
22    \map{
23      \step[fieldsource=journaltitle, match={Physics Letters A},
24        replace={Phys. Lett. A}]%
25    }
26  }
27 }
```

#### 2.10.7 文献表和标注的排序

排序涉及到两个方面的内容，一是文献表中的排序，二是标注的排序。其中标注的排序通常是在同一位置标注多个标签时，这种情况往往是多个引用关键字（词）传递给一个标注（引用）命令。一般情况下标注是不进行排序的，而是以引用时给出的顺序进行排序，当需要进行标签排序时需要启用 sortcites 选项，这时 biblatex 会按照当前文献表中的顺序对标注标签进行排序。

而文献表中的文献排序由 sorting 选项控制，常见的值包括：

<b>nty</b>	按照姓名、标题、年份排序。
<b>nyt</b>	按照姓名、年份、标题排序。
<b>nyvt</b>	按照姓名、年份、卷数、标题排序。
<b>anyt</b>	按照字母标签、姓名、年份、标题排序。
<b>anyvt</b>	按照字母标签、姓名、年份、卷数、标题排序。
<b>ynt</b>	按照年份、姓名、标题排序。
<b>ydnt</b>	按照年份（降序）、姓名、标题排序。
<b>none</b>	不进行排序。所有的条目按照引用顺序处理。
<b>debug</b>	按照条目键值排列。该选项只用于程序调试。
<b>name</b>	使用由 DeclareSortingTemplate 定义的排序模板名 name。



注意：除非另加说明，文献条目总是按升序排列。

通常情况下，在选择使用某种样式时，文献表的排序方式就已经设定。当需要调整排序时，就需要在加载 biblatex 宏包时显式地给出 sorting 选项，并可根据需要选择上述给出的选项值。

另外，因为 biber 能充分利用 perl 的一些模块的功能，所以 biblatex 还能根据 Unicode::Collation::locale 模块提供不同语言的排序的本地化调整方案。对于中文，默认的排序规则是按照中文字符的 unicode 编码进行排序，但可以通过设置选项来进行调整，对于中文有如下调整方案：zh, zh\_\_pinyin, zh\_\_big5han, zh\_\_gb2312han, zh\_\_stroke, zh\_\_zhuyin。

一般的，biblatex 利用 sortlocale 选项将调整方案名传递给 biber，biber 利用 Unicode::Collation::locale 模块提供对应的排序。因此 biblatex 可以设置选项为：

- sortlocale=auto 或者不设置该选项，为默认的 unicode 编码顺序
- sortlocale=zh, 为 unicode 编码顺序
- sortlocale=zh\_\_pinyin, 为拼音顺序
- sortlocale=zh\_\_big5han, 为 big5 编码顺序
- sortlocale=zh\_\_gb2312han, 为 GB-2312 顺序
- sortlocale=zh\_\_stroke, 为笔划数顺序
- sortlocale=zh\_\_zhuyin, 为注音顺序

中文排序本地化调整的测试文档见 [testlocale.tex](#)

当然这种调整可以直接利用在 biber 编译时设置相应的选项来实现，比如需要按拼音或字顺进行排序时，可使用 biber 选项，此时 biblatex 不需要设置 sortlocale 选项。

#### 示例 67 中文文献排序时的 biber 选项

代码

```
1 %按拼音排序, biber命令
2 biber -l zh__pinyin jobname
3 %按笔画排序, biber命令
4 biber -l zh__stroke jobname
```

一个简单的中文作者排序示例见(八) [如何对文献表中的参考文献进行排序?](#)

有的时候，可能希望标注标签的排序和文献表中的顺序不同，操作方法，可以参考 [citation 的顺序和 bibliography 的顺序如何不关联](#)。其中使用了 biblatex 宏包提供参考文献文境的概念，即不同的文境下可以使用不同的排序方式。

### 2.10.8 排序模板和排序的手动调整

前述的排序选项 nyt 等等，都是 DeclareSortingTemplate 定义的排序模板。在排序模板的定义过程中 biblatex 通常是这么考虑的：

首先要考虑的事项总是条目的 presort 域。如果该域没有定义，biblatex 会使用缺省值“mm”作为预排序字符串。

其次考虑的是 sortkey 域。如果该域有定义，它将作为主要的排序关键字。此时除了 presort 域，将不考虑其它信息。如果 sortkey 域没有定义，排序将使用姓名信息。本宏包将依次尝试使用 sortname、author、editor 和 translator 等域。考虑哪些域也取决于 useauthor、useeditor 和 usetranslator 选项的设置。如果这三个选项都没有启用，那么 sortname 也将被忽略。请注意，所有的名称域都与 maxnames 和 minnames 有关。

如果没有名称域是合适的，或者由于它们没有定义、或者由于 use<name> 域都未启用，那么 biblatex 将采用 sorttitle 和 title 作为最后的备选。

余下考虑的诸项依次是: sortyear 域 (如果给出的话), 否则考虑 year 域的前四个数字; sorttitle 域 (如果给出的话), 否则考虑 title 域; volume 域。

上面的这些说明并不全面, 只是 biblatex 在定义类似 nty 排序模板时的考虑, 定义其它模板可能还有其它考虑, 比如 labelalpha, extraalpha 等, 但我们可以得出到一个结论就是 biblatex 通常选择排序内容的时候首先会考虑该部分内的 sort 相关的域。比如排序作者时首先考虑 sortname, 而不是 author, 排序标题时, 首先考虑 sorttitle, 而不是 title。因此我们如果需要对某种排序做调整, 那么就可以手动的调整 sort 相关的域, 即再 bib 文件中增加这些域并设置合适的信息用于排序。这些域包括: presort, sortkey, sortname, sorttitle, sortyear, sortdate 等。需要注意的是, 因为对 bibtex 格式的兼容性, biblatex 也支持域 key, 该域等价于 biblatex 中 sortkey 域, 所以当文献信息中出现 key 域时, 表示要对这些文献进行排序设定。

### 2.10.9 文献信息索引

当需要对参考文献的信息进行索引时, 可以使用 indexing 选项。启用 indexing 后, biblatex 的标准样式默认会将作者和题名加入到索引中去, 如果需要增加一些信息的索引则需要对样式文件做修改。另外, 要生成索引还需要使用索引相关的宏包, 如 makeidx。宏包加载如例 68 所示:

示例 68 文献信息索引

1 见文件 testindexofbib.tex

代码

## 2.11 专题: 多语言文献

### 2.11.1 多语言混合文献表

一般情况下, 在国内不太会超过两种语言混合的情况, 比如中英文混合的情况。但有时可能也会存在多种语言, 比如存在中/英/日/俄这种多语言环境。图 28 给出了这样一个示例, 其中不同的语言使用了不同的本地化字符串。

## 参考文献

- [1] PHILIPPE, B., JEAN, T., PHILIPPE, B., et al. ÔÇLes interprétations de la crise de mai-juinŕ[C]//Revue française de science politique. 1968 :19-46.
- [2] Greenwood, J., Hercowitz, Z., Huffman, G. W., et al. Investment, capacity utilization, and the real business cycle[J]. The American Economic Review, 1988: 402-417.
- [3] Кузнецов, А. В., Кузнецов, С. В., Кузнецов, Д. В., и др. Русский язык и языковая личность[M]. М.: Диалог МГУ, 1998: 28.
- [4] 刘凤良, 章潇萌, 于泽, 等. 高投资、结构失衡与价格指数二元分化[J]. 金融研究, 2017(2): 54-69.
- [5] 青木昌彦, 青木昌一, 青木昌二, 等. メインバンク システムのモニタリング機能としての特徴[M]. 東洋経済新報社, 1996: 288.

图 28 多语言混合文献表

使用 xelatex 编译时下, 由于其原生支持 unicode 的特性, 在 tex 文档内实现多语言混合比较容易实现, 要正确显示关键在于合适的字体设置。一般情况下中/日/韩语利用 ctex 宏包可以解决。而西文如英/法/俄可以利用 fontspec 宏包选择合适的字体来解决。

对于参考文献来说还有一个更重要的问题是本地化字符串问题。中英文情况下, 中文利用在英文本地化文件基础上新增本地化字符串加以解决。但其它语言需要自己的解决方案。

中文环境下的用户, 可以使用 language 域来区分文献的语言类型, 默认情况下该域不需要人工输入, 可由 biber 根据文献信息自动判断, 但也可以手动输入来人工指定。根据 biblatex 提供的多语言解决方案, 还需要利用 langid/langidopt 域, 以及 babel/polyglossia 宏包的支持。

解决方案做如下的考虑:

- (1) 日韩语采用类似中文的方式处理, 即通过在英语本地化文件基础上处理。因此 langid 通常设为 english。对于日文通常作者这种信息都是用的汉字, 因此用中文的字符串。韩语暂没有考虑它自身的字符串, 而使用英文的字符串。当不存在除英文外的其它西语时, langid 也不需要处理。注意: 默认情况下 langid 都是自动处理的, 用户无需过多关注。
- (2) 如俄语/法语这样的西方语言, 通过 biblatex 提供的方案解决。即: 使用 babel/polyglossia 宏包和 langid/langidopt 域。这时如果存在中日韩语, 那么相应文献也需要设置 langid。因此, 需要在 tex 文档内加入 babel 宏包以及需要使用的语言, 需要使用本地化字符串的西语都要加入, 否则无法自动切换。然后在加载 biblatex 宏包时使用 autolang=other 选项。

### 2.11.2 多语言对照的文献表

某些期刊对于参考文献表有双语文献对照的要求, 具体要求见 GB/T 7714-2015 第 6.1 节。这一问题可以通过条目集类型 (set)/或者条目关联 (related) 来解决, 多语言对照的情况与双语对照本质是一样的, 因此下面主要讨论双语对照的文献表。

**2.11.2.1 利用条目集类型满足双语文献要求** 设置条目集类型 (set) 有静态和动态两种方法: 动态方法使用更为方便, 在写文档时候, 直接利用 defbibentryset 设置双语文献的 set, 然后引用 set 的 bibtex 键。比如:

示例 69 设置 set 条目集用于双语文献动态方法

代码

```
1 \defbibentryset{bilangyi2013}{易仕和2013--, Yi2013--}
2 专著, 双语文献引用\cite{bilangyi2013}
```

动态方法的使用更方便, 直接在写文档时候, 将双语文献设置成 set, 然后引用 set 的 bibtex 键。比如:

示例 70 设置 set 条目集用于双语文献动态方法

代码

```
1 %file:egbilang.tex
2 \documentclass{report}
3 \usepackage{ctex}
4 \usepackage{geometry}
5 \geometry{paperwidth=15cm,paperheight=9.5cm,top=0cm, bottom=1.5cm, left=1cm,right=1cm}
6 \usepackage{titlesec}
7 \titleformat{\chapter}{\zihao{5}\heiti}{\thechapter}{1em}{}
8 \titlespacing*{\chapter}{0pt}{0.0\baselineskip}{0.0\baselineskip}{0pt}
```

```

9 \titleformat{\section}{\zihao{-5}\heiti}{\thesection}{1em}{}
10 \titlespacing*{\section}{0pt}{0.0\baselineskip}{0.0\baselineskip}[0pt]
11 \usepackage[backend=biber,style=gb7714-2015]{biblatex}
12 \addbibresource[location=local]{example.bib}
13
14 \begin{document}
15 \chapter{序章}
16 序章内容\cite{GPS1988--}
17 \defbibentryset{bilangyi2013}{易仕和2013--,Yi2013--}
18 \defbibentryset{bilangzhang2007}{张敏莉2007-500-503,Zhang2007-500-503}
19 专著, 双语文献引用\cite{bilangyi2013}\cite{bilangzhang2007}
20 \printbibliography[heading=subbibliography,title=参考文献]
21 \end{document}

```

得到的参考文献打印结果如图29所示。

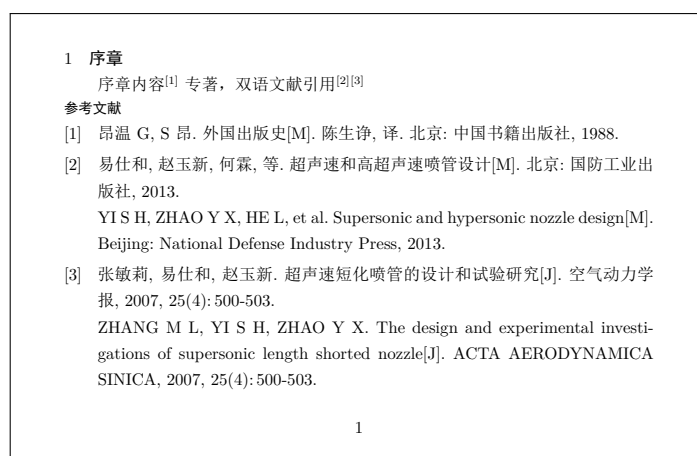


图 29 双语文献参考文献

在 biblatex v3.8 及以上版本中, 因为 set 条目类型除了子条目关键词信息外, 并无其他信息, 因此它的标注标签通常会空的, 这与 biblatex 版本升级有关, biblatex v3.7 及之前版本没有这个问题, 因为这些版本中 set 中带有第一个子条目的信息。目前该问题的解决方法是设置一个指定格式和内容中间无空格无英文逗号的关键词, 用它来作为标签。。比如:

#### 示例 71 设置 set 条目集用于双语文献动态方法

```

1 \defbibentryset{易仕和, 等, 2013}{易仕和2013--,Yi2013--}
2 专著, 双语文献引用\cite{易仕和, 等, 2013}

```

代码

注意到“易仕和, 等, 2013”中的逗号是中文全角逗号, 这样使得“易仕和, 等, 2013”以一个整体字符串当作关键字, 而不会被分开解析。但是这种解决方案中的中文全角逗号与其它标签的英文逗号的差异使得该问题并没有完美解决。这可以用静态方法手动添加标签来解决, 对于 biblatex v3.8 及以上版本还可以利用后面介绍的关联 (related) 方法来解决。

静态方法是在 bib 源文件中给出条目集 (set) 并使用 biber 后端进行解析, 条目的域信息采用如下方法定义:

#### 示例 72 设置 set 条目集用于双语文献静态方法

```

1 @Set{set1,
2 entryset = {key1,key2,key3},

```

代码

```

3 }
4 %如果要达到上例动态设置set一样的结果, 在bib文件中静态设置set条目如下:
5 @Set{bilangyi2013,
6 entryset = {易仕和2013--, Yi2013--},
7 }

```

使用静态条目集时, 如果仅采用上述这般简单设置, 中文排序会出现问题, 条目集会出现在文献表末尾, 这是因为条目集没有设置 `language` 域用于排序, 而通常的条目都会利用动态数据修改设置 `language` 域, 在 `biber` 运行中因为要解析文献集, 所以无法对 `language` 域进行处理。

但使用动态条目集方法没有这一问题, 因为其解析过程直接会利用第一个子条目的排序信息。静态方法的这个问题可以在 `set` 条目中手动设置 `language` 域来修正。此外, 对于 `v3.8` 以上版本的 `biblatex`, 无论动态方法还是静态方法, 条目集不复制第一个子条目信息, 因此引用时也无法生成标注标签, 这就是前面介绍动态方法时讨论过的问题, 该问题也可以通过在 `set` 条目中手动设置 `label` 域来解决。

设置方法为:

#### 示例 73 在 bib 文件中正确设置 set 条目集的静态方法

代码

```

1 %在bib文件中静态设置set条目如下, 其中:
2 %手动设置userb域用于解决排序问题
3 %手动设置label域用于解决标注标签问题
4 @Set{bilangyi2013,
5 entryset = {易仕和2013--, Yi2013--},
6 label={易仕和, 等, 2013},
7 language={chinese}
8 }

```

还要注意: 动态 `set` 条目集等价于使用 `nocite` 命令, 因此只要定义了动态条目集的文献都会出现在文献表中, 因此如果不引用相应的文献, 那么就不应该对其定义动态条目集。

**2.11.2.2 利用条目关联满足双语文献要求** 除上述给出的条目集方案外, 关联条目方法则是另一种可行方案, 该方案的讨论可以见“Again about the `set label for authoryear style`”<sup>8</sup>。该方案同样也有静态和动态两种方法, 静态就是修改 `bib` 文件内容, 动态则是在 `tex` 源文档中做设置。

静态方法很简单, `bib` 文件中条目设置如例74所示, 它能解决双语同时显示的问题, 也能解决排序和标注标签问题, 唯一的问题在于修改了 `bib` 文件后当不需要双语文献时, 它还需要改回来, 这会带来不便, 因此可以考虑下面的动态方法, 但要注意动态方法需要利用多个 `DeclareStyleSourcemap`, 因此该方法只适用于 `biblatex v3.7` 及以上版本。

#### 示例 74 在 bib 文件中正确设置关联条目的静态方法

代码

```

1 %在bib文件中静态设置条目如下, 其中:
2 %易仕和2013--条目中增加了related域用于关联其对应的英文条目Yi2013--
3 @Book{易仕和2013--,
4 Title = {超声速和高超声速喷管设计},
5 Address = {北京},
6 Author = {易仕和 and 赵玉新 and 何霖 and 张敏莉},
7 Publisher = {国防工业出版社},

```

<sup>8</sup><https://github.com/plk/biblatex/issues/681>



```

8   Year = {2013}
9   related = {Yi2013--}
10 }
11 @Book{Yi2013--,
12   Title = {Supersonic and hypersonic nozzle design},
13   Address = {BeiJing},
14   Author = {Yi, S H and Zhao, Y X and He, L and Zhang, M L},
15   Publisher = {National Defense Industry Press},
16   Year = {2013}
17 }

```

动态方法利用动态数据修改自动添加 related 域，避免对 bib 文件做直接修改。本样式中对该过程进行了封装，定义一个新的命令 `defdoublelangentry`，例如：

示例 75 设置关联条目的动态方法

```

1 \defdoublelangentry{易仕和2013--}{Yi2013--}

```

代码

使用该命令后，可以引用主条目“易仕和 2013-”生成双语文献。但要注意由于 `DeclareStyleSourcemap` 命令只能在导言区中使用，因此 `defdoublelangentry` 命令也只能出现在导言区中，这也是相比条目集动态方法的唯一遗憾。

双语对照文献的两种动态方法基于 `set` 和 `related` 的方法测试，参见：[bibtex-gb7714-2015 宏包手册](#)。

### 2.11.3 关联与条目集的差异

在 `bibtex` 中不同的文献之间产生一定的联系的方式共有三种，分别是交叉引用、关联、集合。其中交叉引用使用 `crossref`，详见 2.10.6 节，这是传统的 `bibtex` 使用的方式，主要用于数据的继承。而关联 (`related`) 和条目集合 (`set`) 是则是两种不同条目信息集合输出的方式。

关联主要是要体现出不同条目之间的关系，比如“再版”，“原版”，“重印”，“多卷”，“摘录于”等等。`bibtex` 通过使用条目域 `related`、`relatedtype` 和 `relatedstring` 来实现这种一般性的机制，通过 `related` 域来关联不同的条目，通过 `relatedtype` 制定关联条目间的关系的本地化字符串名，比如：`reprintof` 等。而 `relatedstring` 域可以用于覆盖那些通过 `relatedtype` 确定的本地化字符串，此时 `relatedtype` 域给出的信息用于确定该字符串的格式。`bibtex` 关联条目的例子详见 `bibtex` 附带的 `90-related-entries-biber.pdf`。`bibtex` 还提供 `relatedoptions` 域用于设置一些选项，选择值有 `dataonly`，`skiplab`，`skipbib`，`skipbiblist` 等，设置该选项的目标主要是为了使用默认 `dataonly` 选项所没有的功能，该域一般情况下不常用，使用时遇到问题可以在 `github` 咨询 `bibtex` 的开发和维护者。

而条目集是用单个引用并在参考文献中作为一项列出的一组条目，主要的概念是多个条目集合为一个分组并输出。它在条目输出上与关联是类似的，差别在于条目间没有更明确的关系存在。如前所述，条目集可以有两种手动构成方式，一种静态，一种动态。除此之外还有一种更方便的使用方式就是利用 `mcite` 命令，该命令会自动构造动态条目集。`mcite` 类的命令指示对 `cite` 等命令的封装，因此可以有 `mcite`，`Mcite`，`mparencite`，`Mparencite`，`mfootcite`，`mfootcitetext`，`mtextcite`，`Mtextcite`，`msupercite` 等命令。其原理如例所示：

示例 76 设置 `set` 动态条目集的 `mcite` 方法

```

1 \mcite*[pre][post]{setA,*keyA1,*keyA2,*keyA3}
2
3 %等价于执行
4
5 \defbibentryset{setA}{keyA1,keyA2,keyA3}%
6 \cite*[pre][post]{setA}

```

代码



当然要注意, 要使用 `mcite` 类命令, 要在加载 `biblatex` 时启用 `mcite` 模块选项。

需要注意的是条目集在有标签的文献样式中使用更为合适, 比如 `numeric`, `alpha` 类样式, 而在以文献信息本身作为文献不同区分的样式中并不常用, 当然这是西方用户的习惯。中文环境内因为很多时候有双语对照的需求, 那么再不适合也要用。当然对于多语言对照的情况, 在使用较新版本的 `biblatex` 时, 更推荐使用关联的方法。

## 2.12 专题: 脚注题注等环境中的引用和文献表

文献不仅可以在正文中引用, 也可以在脚注, 题注, 小页环境中引用, 下面分别进行说明:

### 2.12.1 脚注中的引用

要在脚注中的引用文献, 直接在 `footnote` 脚注的内容中添加 `cite` 等引用命令即可, 比如:

示例 77 脚注中引用参考文献

代码

```
1 %file:egciteinfoot.tex
2 \documentclass{report}
3 \usepackage{ctex}
4 \usepackage{geometry}
5 \geometry{paperwidth=12cm,paperheight=8cm,top=0cm, bottom=1.5cm, left=1cm,right=1cm}
6 \usepackage{hyperref}
7 \usepackage{titlesec}
8 \titleformat{\chapter}{\zihao{5}\heiti}{\thechapter}{1em}{}
9 \titlespacing*{\chapter}{0pt}{0.5\baselineskip}{0.5\baselineskip}[0pt]
10 \titleformat{\section}{\zihao{-5}\heiti}{\thesection}{1em}{}
11 \titlespacing*{\section}{0pt}{0.5\baselineskip}{0.5\baselineskip}[0pt]
12 \usepackage[backend=biber,style=gb7714-2015]{biblatex}
13 \addbibresource[location=local]{example.bib}
14
15 \begin{document}
16 \chapter{序章}
17 正文内容脚注中引用参考文献\footnote{biblatex使用可以参考宏包手册\cite{Lehman2015}}
18 \printbibliography[heading=subbibliography,title=本章参考文献]
19 \end{document}
```

结果如图30所示。

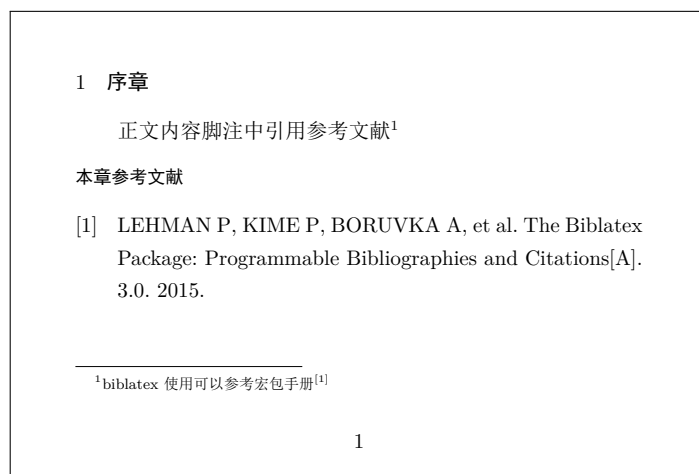


图 30 脚注中引用参考文献

### 2.12.2 题注中的引用

在题注中引用文献方法是类似的，直接在 caption 题注的内容中添加 cite 等引用命令即可，比如：

示例 78 题注中引用参考文献

代码

```

1 %file:egciteincaption.tex
2 \documentclass{report}
3 \usepackage{ctex}
4 \usepackage{geometry}
5 \geometry{paperwidth=12cm,paperheight=9cm,top=0cm, bottom=1.5cm, left=1cm,right=1cm}
6 \usepackage{titlesec}
7 \titleformat{\chapter}{\zihao{5}\heiti}{\thechapter}{1em}{}
8 \titlespacing*{\chapter}{0pt}{0.5\baselineskip}{0.5\baselineskip}[0pt]
9 \titleformat{\section}{\zihao{-5}\heiti}{\thesection}{1em}{}
10 \titlespacing*{\section}{0pt}{0.5\baselineskip}{0.5\baselineskip}[0pt]
11 \usepackage{hyperref}
12 \usepackage[backend=biber,style=gb7714-2015]{biblatex}
13 \addbibresource[location=local]{example.bib}
14
15 \begin{document}
16 \chapter{序章}
17 \begin{figure}[!htb]
18   \centering
19   \fbox{\parbox[c][2cm][c]{4cm}{\centering\large example figure}}
20   \caption{在题注中引用文献\cite{GPS1988--}}\label{cite:incaption}
21 \end{figure}
22 \printbibliography[heading=subbibliography,title=本章参考文献]
23 \end{document}

```

结果如图31所示。

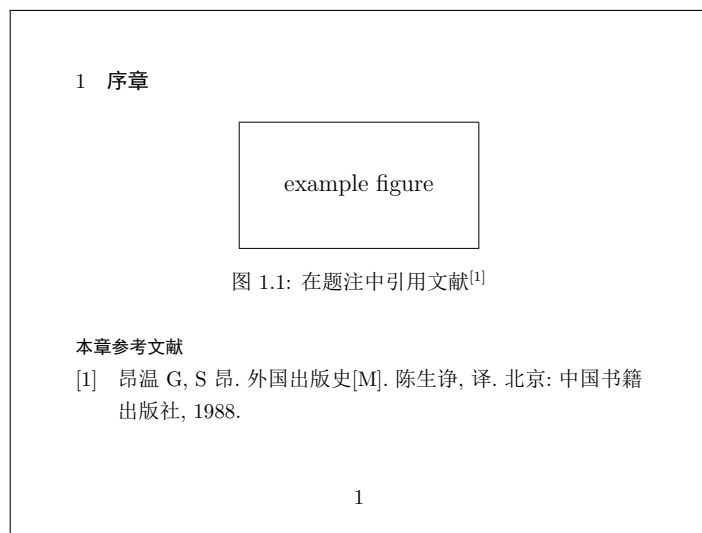


图 31 题注中引用参考文献

### 2.12.3 小页环境中的引用

在小页环境中引用文献方法也是类似的，直接在小页环境的内容中添加 cite 等引用命令即可，比如：

## 示例 79 小页环境中引用参考文献

代码

```

1 %file:egciteinminipage.tex
2 \documentclass{report}
3 \usepackage{ctex}
4 \usepackage{geometry}
5 \geometry{paperwidth=12cm,paperheight=6cm,top=0cm, bottom=1.5cm, left=1cm,right=1cm}
6 \usepackage{titlesec}
7 \titleformat{\chapter}{\zihao{5}\heiti}{\thechapter}{1em}{}
8 \titlespacing*{\chapter}{0pt}{0.5\baselineskip}{0.5\baselineskip}[0pt]
9 \titleformat{\section}{\zihao{-5}\heiti}{\thesection}{1em}{}
10 \titlespacing*{\section}{0pt}{0.5\baselineskip}{0.5\baselineskip}[0pt]
11 \usepackage{hyperref}
12 \usepackage[backend=biber,style=gb7714-2015]{biblatex}
13 \addbibresource[location=local]{example.bib}
14
15 \begin{document}
16 \chapter{序章}
17 \fbox{
18 \begin{minipage}{0.5\linewidth}\centering
19 使用小页环境引用文献\cite{GPS1988--}
20 \end{minipage}}
21 \printbibliography[heading=subbibliography,title=本章参考文献]
22 \end{document}

```

结果如图32所示。

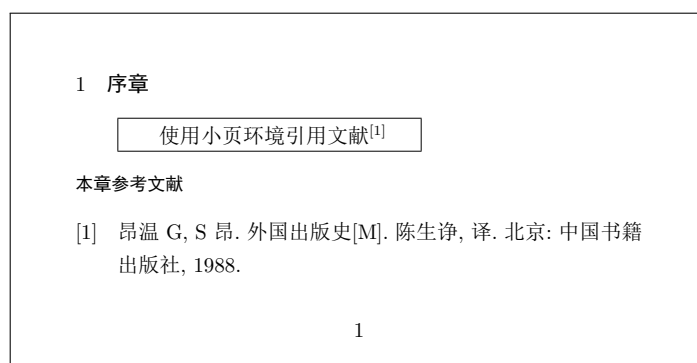


图 32 小页环境中引用参考文献

#### 2.12.4 使用 biblatex 命令实现脚注文献表

前面说过, biblatex 可以访问和利用参考文献源文件中的任何数据。其中, 就可以利用这一特点形成一个相关功能即: 将参考文献表放到脚注中, 这在某些需要的时候是很有用的。

biblatex 提供了一个 footfullcite 命令实现将文献表放到脚注中, 看例80给出的代码, 3 个 footfullcite 命令将 3 条文献放入脚注中, 文献的著录格式仍然是指定的 gb7714-2015ay 样式。

#### 示例 80 脚注参考文献表

代码

```

1 %file:egbibinfoot.tex
2 \documentclass{article}
3 \usepackage{ctex}
4 \usepackage[paperwidth=12cm,paperheight=9cm,%
5 left=1cm,right=1cm,top=1cm,bottom=1.5cm]{geometry}
6 \usepackage[colorlinks=true,pdfstartview=FitH,linkcolor=blue,
7 anchorcolor=violet,citecolor=magenta]{hyperref}%书签功能, 选项去掉链接红色方框
8 \usepackage{titleref}%标题引用

```



```

15 % stable, % Make footnotes stable in section titles
16 perpage, % Reset on each page
17 % %para, % Place footnotes side by side of in one paragraph.
18 side, % Place footnotes in the margin
19 % ragged, % Use RaggedRight
20 % marginal,
21 % norule, % suppress rule above footnotes
22 % %hang,
23 % multiple, % rearrange multiple footnotes intelligent in the text.
24 % %symbol, % use symbols instead of numbers
25 ]{footmisc}
26
27 \begin{document}
28
29 \section{页边的文献表}
30 参考文献\cite{白书农1998-146-163}
31 \footfullcite{陈志勇2011--}
32 \footfullcite{储大同2010-721-724}
33 \footfullcite{顾炎武1982--}
34 中文段落参考文献中文段落参考文献中文段落参考文献中文段落参考文献中文段落
35 参考文献中文段落参考文献中文段落参考文献中文段落参考文献中文段落参考文献
36 中文段落参考文献中文段落参考文献中文段落参考文献
37
38 \newpage
39 \newgeometry{left=1cm,right=1cm,top=1cm,bottom=2cm}
40 \printbibliography[title=【全部引文】]
41 \end{document}

```

结果如图34所示。

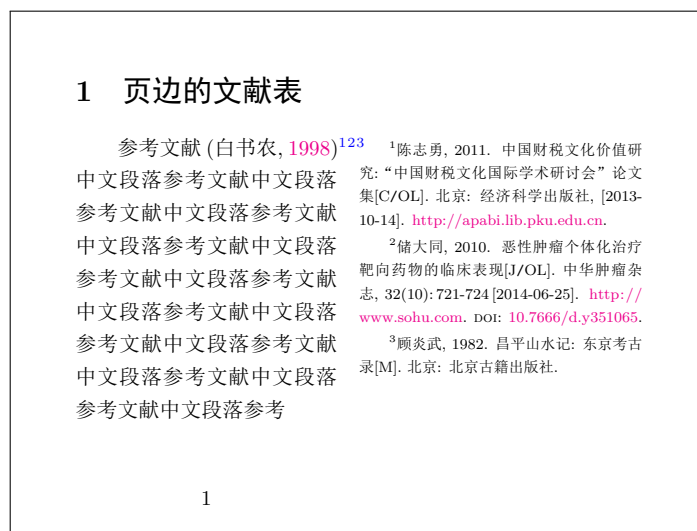


图 34 旁注中的参考文献表

### 2.12.6 使用自定义环境实现旁注文献表

旁注的文献表还可以通过自定义参考文献环境命令的方式实现，这种方式下可以同时存在脚注和旁注参考文献。例82给出了测试代码，这也是参考文献表环境定义的一个例子。需要注意，这里定义的命令中使用了 `keyword` 域中的引用关键词信息，这一信息是由 `gb7714-2015` 样式文件处理后提供的，所以使用这种方式必须加载 `gb7714-2015` 样式。而且这里的环境定义相对比较简单，可以进一步改进。或者也可以利用 `biblatex` 提供的接口使用文献的条目类型驱动，具体可以参考 `biblatex` 宏包说明的 `Mixing Programming Interfaces` 一节。

示例 82 另一种脚注和旁注参考文献表

代码

```

1 %file:egbibinfootmargin
2 \documentclass[twoside]{article}
3 \usepackage{ctex}
4 \usepackage{geometry}
5 \geometry{includemp,paperwidth=21cm,paperheight=19cm,%
6 left=1cm,right=1cm,marginparwidth=6cm,top=1cm,bottom=1.5cm}
7 \usepackage{xcolor}
8 %书签功能, 选项去掉链接红色方框
9 \usepackage[CJKbookmarks,colorlinks,bookmarksnumbered=true,pdfstartview=FitH,linkcolor=blue]{hyperref}
10 \usepackage[backend=biber,style=gb7714-2015]{biblatex}
11 \addbibresource[location=local]{example.bib}
12
13 %\newcommand{\itemcmd}[1]{#1\par}
14 \defbibenvironment{marginref}{\begin{group}}
15 {\end{group}}{\zihao{6}\songti}{\newline\itemcmd}
16 \defbibheading{marginref}{}
17 \newcommand{\pz}[1]{% 定义 pz 为旁注命令
18 \marginpar[\flushright\textcolor{blue}{\scriptsize#1}]{\scriptsize#1}}
19 \newcommand{\pzcite}[1]{%
20 \cite{#1}
21 \marginpar[\flushright\textcolor{blue}{{\printbibliography[env=marginref,keyword=#1,heading=marginref]}}]}
22 {\printbibliography[env=marginref,keyword=#1,heading=marginref]}}
23
24 \begin{document}
25 \section{脚注和旁注中的文献表}
26 参考文献\pzcite{Eggers--}中文段落参考文献中文段落参考文献中文段
27 参考文献中文段落参考文献中文段落参考文献中文段落参考文献中文段落
28 参考文献中文段落参考文献中文段落参考文献中文段落参考文献中文段落
29 参考文献\pzcite{汤万金2013-09-30--}中文段落参考文献中文段落参考
30 参考文献\footfullcite{刘裕国2013-01-12--}
31 \footfullcite{Dublin2012-06-14--}
32 \footfullcite{王夫之1845--}
33 中文段落参考文献中文段落参考文献中文段落参考文献中文段落
34 参考文献中文段落参考文献中文段落参考文献中文段落参考文献中文段落
35
36 text text text text text text text text
37 \fbox{here is a marginpar}\pz{some note.}
38 text text text text text text text text
39 text text text text text text text text
40
41 参考文献中文段落参考文献中文段落参考文献中文
42 \fbox{这里是旁注}\pz{中文旁注}
43 段落参考文献中文段落参考文献中文段落参考文献
44 中文段落参考文献中文段落参考文献
45 \printbibliography[title=【参考文献】]
46 \end{document}

```

结果如图35所示。

## 2.13 专题: beamer 类中的参考文献

在 beamer 类中使用 biblatex 生成参考文献与一般文档类中大体上一致, 略有差别, 主要是 refsection 环境无法使用。在实际应用中, beamer 类有两种文献表是主要的, 一是文末的文献表, 二是脚注中的文献表。





```

17 \renewcommand{\footnotesize}{\tiny}
18
19 \begin{document}
20 \begin{frame}[plain]
21   \titlepage
22 \end{frame}
23
24 \begin{frame}{测试参考文献}
25   \tiny
26   在脚注中引用或者把文献表放到脚注中
27   \footnote{在脚注中引用\footcite{Saito2006-169-176}}
28   \footfullcite{中国职工教育研究会1985--}
29   \footfullcite{Fontana2002-309-313}
30   \footfullcite{Robertson2011--}
31   \footfullcite{雷光春2012--}
32   \footfullcite{Humphrey1971--}
33   \footfullcite{马欢2011-27-27}
34   \footfullcite{中国图书馆学会1957--}
35   \footfullcite{刘彻东1998-38-39}
36 \end{frame}
37
38 \begin{frame}{参考文献}
39 \printbibliography[heading=bibliography,title=参考文献]
40 \end{frame}
41 \end{document}

```



图 36 beamer 类脚注中的参考文献表

### 2.13.2 文献表中的条目序号

因为 beamer 类中重定义了一些内容，比如空格，list 环境的标签等，或者因为一些非标准标签的块环境自定义，biblatex 默认使用的文献表环境，可能会导致文献表的序号丢失 (图38中文献表只有第一条有序号) 或者不是想要的顺序编码的标签 (图37中的文献表就是一个图标，而不是一个序号，这也许不是想要的)。这个序号丢失的问题来自 ctex 论坛的提问<sup>[17]</sup>。

这种情况下，要得到参考文献表的顺序编码序号，有两种方法：一是使用自定义的参考文献环境，避免使用基于 list 的环境，二是将 list 环境命令的标签重新设置回来。第一种方法比如利用 gb7714-2015 样式文件，设置其 align 选项为 gb7714-2015 就可以得到顺序标签，代码如例84所示，结果如图39所示。第二种方法，最方便的是设置 beamer 的内部主题，比如



图 37 beamer 类全局参考文献表

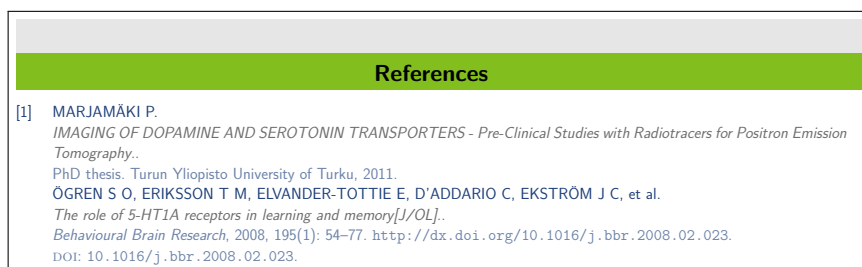


图 38 一个自定义 block 中参考文献表 (序号丢失)

defbeamertemplate\*{bibliography item} 和 setbeamertemplate{bibliography item}, 代码如例85所示, 结果如图40所示, 其中利用 setbeamertemplate{bibliography item} 命令也可以设置成其他标签形式, beamer 标准主题<sup>[18]96</sup> 提供了 default, article, book, online, triangle, text 等 6 种形式, 当不想要例40中顺序编码标签时, 完全可以选择这些标签形式。

示例 84 beamer 中 biblatex 使用 gb7714-2015 定义的

align=gb7714-2015 选项

代码

```
1 %file:egbeamersn.tex
2 \usepackage[backend=biber,bibstyle=gb7714-2015-m,%nature,%
3 citestyle=gb7714-2015,%backref=true%gb, gbt7714_2005_n,
4 align=gb7714-2015%
5 ]{biblatex}%biber, caspervector
6 \renewcommand{\bibfont}{\footnotesize}
7 \addbibresource[location=local]{bib.bib}
```

示例 85 beamer 中 biblatex 利用内部主题设置文献表标签

代码

```
1 %file:egbeamersnw.tex
2 \usepackage[backend=biber,bibstyle=gb7714-2015-m,%nature,%
3 citestyle=gb7714-2015,%backref=true%gb, gbt7714_2005_n,
4 align=left%
5 ]{biblatex}%biber, caspervector
6 \renewcommand{\bibfont}{\footnotesize}
7 \addbibresource[location=local]{bib.bib}
8 \defbeamertemplate*{bibliography item}{numeric}
9 {\hbox{[\printfield{labelnumber}]}}
10 \setbeamertemplate{bibliography item}[numeric]%book,numeric
```

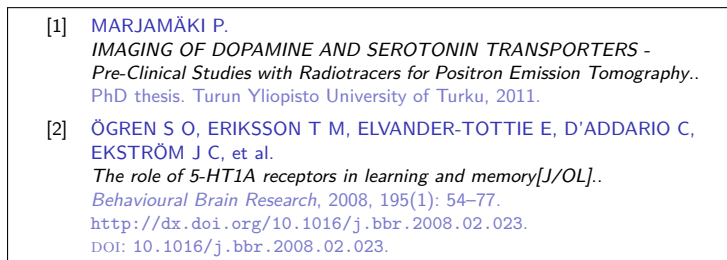
- 
- [1] MARJAMÄKI P.  
*IMAGING OF DOPAMINE AND SEROTONIN TRANSPORTERS - Pre-Clinical Studies with Radiotracers for Positron Emission Tomography..*  
PhD thesis. Turun Yliopisto University of Turku, 2011.
- [2] ÖGREN S O, ERIKSSON T M, ELVANDER-TOTTIE E, D'ADDARIO C, EKSTRÖM J C, et al.  
*The role of 5-HT1A receptors in learning and memory[J/OL]..*  
*Behavioural Brain Research*, 2008, 195(1): 54–77.  
<http://dx.doi.org/10.1016/j.bbr.2008.02.023>.  
DOI: 10.1016/j.bbr.2008.02.023.

图 39 beamer 中 biblatex 使用 gb7714-2015 定义的  
align=gb7714-2015 选项后的文献表

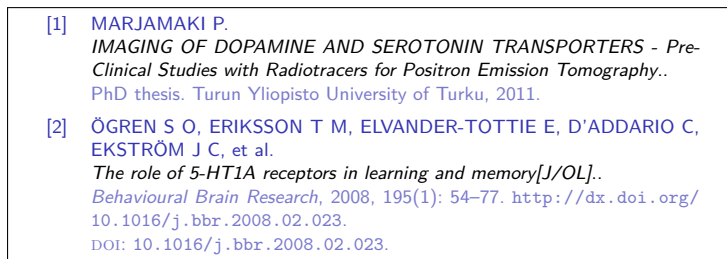
- 
- [1] MARJAMAKI P.  
*IMAGING OF DOPAMINE AND SEROTONIN TRANSPORTERS - Pre-Clinical Studies with Radiotracers for Positron Emission Tomography..*  
PhD thesis. Turun Yliopisto University of Turku, 2011.
- [2] ÖGREN S O, ERIKSSON T M, ELVANDER-TOTTIE E, D'ADDARIO C, EKSTRÖM J C, et al.  
*The role of 5-HT1A receptors in learning and memory[J/OL]..*  
*Behavioural Brain Research*, 2008, 195(1): 54–77. <http://dx.doi.org/10.1016/j.bbr.2008.02.023>.  
DOI: 10.1016/j.bbr.2008.02.023.

图 40 beamer 中 biblatex 利用内部主题设置文献表标签后的文献表

但需要注意的是，第二种方法对于标准 beamer 类或符合其规范的环境下才有效，对于一些自定义的环境可能无法起到作用，比如在 `egpostersnw.tex` 文件中的自定义的 `myblock` 环境中就没有效果。因此只能采用第一种方法，具体见 `egpostersn.tex` 文件的内容。其结果如图 41 所示。

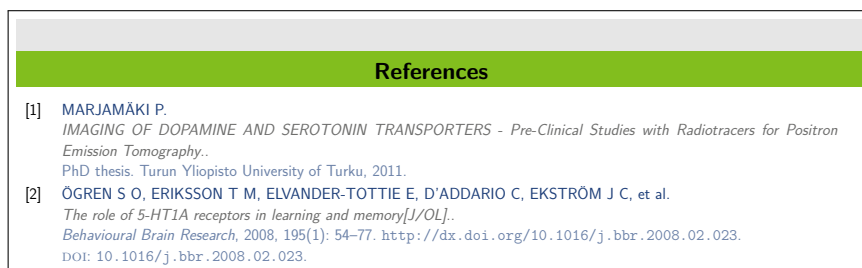
- 
- [1] MARJAMÄKI P.  
*IMAGING OF DOPAMINE AND SEROTONIN TRANSPORTERS - Pre-Clinical Studies with Radiotracers for Positron Emission Tomography..*  
PhD thesis. Turun Yliopisto University of Turku, 2011.
- [2] ÖGREN S O, ERIKSSON T M, ELVANDER-TOTTIE E, D'ADDARIO C, EKSTRÖM J C, et al.  
*The role of 5-HT1A receptors in learning and memory[J/OL]..*  
*Behavioural Brain Research*, 2008, 195(1): 54–77. <http://dx.doi.org/10.1016/j.bbr.2008.02.023>.  
DOI: 10.1016/j.bbr.2008.02.023.

图 41 自定义 block 中使用 gb7714-2015 定义的 align=gb7714-2015  
选项后的文献表

观察 `egpostersnr.tex` 文件所使用的主题 `beamerthemeZH.sty`，理论上 `beamer` 中定义 block 应该不会导致序号丢失的，但为什么这里就丢失了呢？仔细查看后，发现其中使用了 `ragged2e` 宏包中的 `justify` 命令，但事实上 `ragged2e` 说明文档中并没有该命令，而只有 `justifying` 命令，所以序号丢失的问题我估计是主题作者的 `justifying` 命令笔误了，导致了一些未知的问题。事实上只要把这个命令改回来，就不存在丢失的问题了，那么第二种方法的应用自然也没有问题。详见 `egpostersnr.tex`, `egpostersnr.pdf`, `beamerthemeZHM.sty` 文件，其中 `beamerthemeZHM.sty` 就是原 `beamerthemeZH.sty` 文件的修正。

### 2.13.3 beamer 中 biblatex 参考文献表内容格式

参考文献表的内容格式（包括段落、字体和颜色等）的修改，除了可以利用 `biblatex` 提供的钩子（见 2.8 节）、自定义参考文献环境（见 2.8 节）、设计和使用特定的参考文献样式（见 2.6 节）外，还可以利用 `beamer` 主题进行修改，可以从 `Template/Color/Font` 三个角度进行修改。

从实践看，因为 `biblatex` 中给出的各条目参考文献中各块各单元都是一个编组，且这些编组都有其相应的域格式控制，因此段落格式（包括标点）和字体是不能通过 `beamer` 主题修

改的, 而因为 biblatex 的各单元的域格式并没有对颜色进行设置, 因此 beamer 主题可以修改其颜色, 代码如例 86 所示, 结果如图 42 所示。

这里格式添加的原理是在条目开始时添加一个设置信息, 遇到一个 block 时又添加一个设置信息, 因为 biblatex 没有对域格式进行颜色设置, 所以在各单元的编组外的颜色设置信息能够起到作用, 而字体和标点设置信息因为编组内就有, 编组外的设置信息自然无法起到作用。这里的遇到 block 产生添加信息也无非两条路子, 一是重定义 newblock 命令, 二是使用 appto 等命令在其后或前添加命令。

示例 86 beamer 中 biblatex 利用颜色主题设置文献颜色

代码

```

1 %file:egbeamer.tex
2 \usepackage[backend=biber,style=gb7714-2015%gb7714-2015,%nature,%
3 %,backref=true%gb, gbt7714_2005_n,
4 ]{biblatex}%biber, caspervector
5 \renewcommand{\bibfont}{\footnotesize}
6 \addbibresource[location=local]{bib.bib}
7 %颜色设置-有效
8 \setbeamercolor{bibliography item}{fg=yellow}
9 \setbeamercolor{bibliography entry author}{fg=red}
10 \setbeamercolor{bibliography entry title}{fg=green}
11 \setbeamercolor{bibliography entry location}{fg=blue}
12 %\setbeamercolor{bibliography entry note}{fg=pink}
13 \setbeamertemplate{bibliography item}[book]
14 %段落格式设置-无效
15 %\defbeamertemplate*{bibliography entry article}{default}{}
16 %\defbeamertemplate*{bibliography entry title}{default}{\par}
17 %\defbeamertemplate*{bibliography entry location}{default}{\par}
18 %\defbeamertemplate*{bibliography entry note}{default}{\par}
19 \setbeamertemplate{bibliography entry article}[default]
20 \setbeamertemplate{bibliography entry title}[default]
21 \setbeamertemplate{bibliography entry location}[default]
22 \setbeamertemplate{bibliography entry note}[default]
23 %\setbeamertemplate{bibliography entry article}{\newline}
24 %\setbeamertemplate{bibliography entry title}{\newline}
25 %\setbeamertemplate{bibliography entry location}{\newline}
26 %\setbeamertemplate{bibliography entry note}{\newline}
27 %字体设置-无效
28 \setbeamerfont{bibliography item}{family=\rmfamily,size=\small,shape=\itshape}
29 \setbeamertemplate{bibliography entry article}{shape=\itshape}
30 \setbeamertemplate{bibliography entry title}{shape=\itshape}
31 \setbeamertemplate{bibliography entry location}{shape=\itshape}
32 \setbeamertemplate{bibliography entry note}{shape=\itshape}

```



MARJAMÄKI P. IMAGING OF DOPAMINE AND SEROTONIN TRANSPORTERS - Pre-Clinical Studies with Radiotracers for Positron Emission Tomography[D]. Turun Yliopisto University of Turku, 2011.



ÖGREN S O, ERIKSSON T M, ELVANDER-TOTTIE E, et al. The role of 5-HT1A receptors in learning and memory[J/OL]. Behavioural Brain Research, 2008, 195(1): 54-77. <http://dx.doi.org/10.1016/j.bbr.2008.02.023>. DOI: 10.1016/j.bbr.2008.02.023.

图 42 beamer 中 biblatex 利用颜色主题设置文献颜色

### 3 参考文献样式定制方法

biblatex 强大还体现在参考文献样式的定制功能上。本文作为 biblatex 的使用教程,除了在上一节介绍 biblatex 基本的功能和机制(包括:通过选项设置修改、通过简单语句操作修改)外,也将介绍更深层次的功能,主要放在本节中,通过介绍 biblatex-gb7714-2015 样式包等的设计与实现来阐述 biblatex 的高阶功能和强大机制。介绍的步骤,首先是根据样式设计的需求,提出可能要解决的问题,然后分析其原理性思路,接着介绍功能应用,最后则是代码的实现。

#### 3.1 gb7714-2015 样式设计与实现

##### 3.1.1 动机

需求分析是设计的根本来源。之所以要开发 biblatex-gb7714-2015 样式包肯定存在一定的动机,包括填补空白,改进完善等等。具体到细节上,可能包括如下方面,这既是 biblatex-gb7714-2015 样式设计的来源,也是要解决的问题:

- (1) biblatex 版本的兼容性(原来用李志奇分享的样式,但在 biblatex 升级后不能用了,所以希望样式能够克服 biblatex 版本变化带来的问题)
- (2) 格式修改的可逆性(希望样式中做的修改时可以通过选项等方式调整的,以增加用户使用的灵活性,甚至可以通过选项等完全回复到标准样式。)
- (3) 多语言对照(希望实现标准提出的要求)
- (4) 文献表的整体格式(希望实现类似 word 中的标签形式,而不仅是 latex 中 list 的那种风格)
- (5) 新增加的条目类型(除了利用常见条目实现外,还可以针对国标的条目增加新的类型)
- (6) 自定义的域(因为国标有一些文献类型载体标识等的要求,所以要使用自定义的域)
- (7) 著录条目中的作者格式(除提供标准要求格式外,希望提供多种格式选择)
- (8) 著录条目中中英文不同的缩略词(希望实现标准的要求)
- (9) 著录条目中的日期格式(希望实现标准的要求)
- (10) 著录条目中的标点格式(希望实现标准的要求)
- (11) 著录条目中的域的解析(起止卷期等)(希望实现标准的要求)
- (12) 标注的格式(希望实现标准的要求)

上述这些是后期不断完善提出来的总结,其实在初期设计时,主要还是针对目标即 GB/T 7714-2015 标准来一条条的实现,缺什么增加什么,哪个有问题改什么。

##### 3.1.2 著录样式的设计途径

biblatex 的核心之一——参考文献样式文件的设计遵循一定的规则:

- (1) driver 的作用: 条目驱动,修改一些顺序,略去一些输出和标点
- (2) micro 的作用: 输出宏,修改需要的输出内容
- (3) format 的作用: 域打印格式,修改一些斜体,强调样式等。
- (4) command 的作用: 一些设置和命令
- (5) biblatex.sty/def 和各 bbx/cbx 文件中的代码

可以从上述这些方面来设计实现需要的格式要求。

下面通过一个问题查找过程来进一步说明文献著录表的内容组织原理:



**3.1.2.1 文献著录表的内容组织原理** biblatex 是利用 tex 语句控制文献内容组织, 因此我们可以坚定一点就是, 对于文献内容的组织一定不是神秘的, 是可以分析清楚的, 当出现问题的时候, 通过一步步的分析是能解决好的。

如前所述, 根据宏包设置, 使用 biber 解析完参考文献信息后, 就需要利用样式中的各种定义来输出文献表。我们从某一条参考文献来看问题。某一类型的参考文献输出由该类型的 driver 控制, driver 中控制各种 bibmacro 来组织内容输出, bibmacro 利用各种 fieldformat, command 来控制域的格式和标点, 当出现一些特殊的问题时, 我们可以顺着这样的思路来查找。

下面我们举例来说明本样式的作者年制中, 当作者缺省时, 利用 title 判断中英文出错的问题。这个问题本质上由于前面说过的, 当作者缺省时, 把 title 作为标签后, 会把 title 域清除, 因此再利用 title 域信息判断中英文, 自然无法正常工作。我们来看看这是怎么一个过程:

这是一个作者缺省的 book 条目, 在 gb7714-2015ay 样式不使用佚名方式的情况下报错, 主要是 testCJKfirst 函数 (见 3.1.4.1 中的代码) 报错。这个函数主要用在作者, 出版项等这些信息输出时的中英文判断。因为作者缺省, 因此作者输出时就没有判断, 而出错只能是在出版项中。我们把 book 的 driver 中 macro 输出一项一项注释掉并测试发现, 当作者输出相关的 macro 存在时存在错误, 注释掉则没有问题。于是深入来分析:

在 book 类的 driver 简化为如下三项输出:

```
\DeclareBibliographyDriver{book}{%
  \usebibmacro{bibindex}%
  \usebibmacro{begentry}%
  \usebibmacro{author/editor+others/translator+others}%
  \setunit{\labelnamepunct}\newblock
  \usebibmacro{maintitle+title}%
  \newunit
  \usebibmacro{publisher+location+date}%
  \newunit\newblock
  \usebibmacro{finentry}}
```

其中 usebibmacro{author/editor+others/translator+others} 导致出错, 该 macro 的思路是当存在 author 时输出 author, 没有时判断是否存在 editor, 存在则输出, 不存在则判断是否存在 translator, 并进一步判断输出:

```
\newbibmacro*{author/editor+others/translator+others}{%
  \ifboolexpr{
    test \ifuseauthor
    and
    not test {\ifnameundef{author}}
  }
  {\usebibmacro{author}}
  {\ifboolexpr{
    test \ifuseeditor
    and
    not test {\ifnameundef{editor}}
  }
    {\usebibmacro{editor+others}}
    {\usebibmacro{translator+others}}}}
```

显然这个例子中不存在 author 和 editor, 因此宏转到 macro{translator+others}, 我们来看这个宏:

```
\newbibmacro*{translator+others}{%
```

```

\ifboolexpr{
  test \ifusetranslator
  and
  not test {\ifnameundef{translator}}
}
{
  \printnames{translator}%
  \setunit{\printdelim{translator}typedelim}}%
  \usebibmacro{translator+othersstrg}%
  \clearname{translator}}
{}}

```

这个宏来自 biblatex.def, 显然这个宏表明当 translator 不存在则不做任何事情, 这种情况下是不会出错的, 于是我们再找, 发现这个宏在 authoryear.bbx 中重定义了。

```

\renewbibmacro*{translator+others}{%
  \usebibmacro{bbx:translator}{translator+othersstrg}}

\newbibmacro*{bbx:translator}[1]{%
  \ifboolexpr{
    test \ifusetranslator
    and
    not test {\ifnameundef{translator}}
  }
  {
    ...code for existed translator...%
    {\global\undef\bbx@lasthash
      \usebibmacro{labeltitle}%
      \setunit*{\printdelim{nonameyeardelim}}}%
    \usebibmacro{date+extradate}}
}

```

显然当 translator 不存在时, 使用了 labeltitle 宏, 再来看这个宏:

```

\newbibmacro*{labeltitle}{%
  \iffieldundef{label}
  {\iffieldundef{shorttitle}
    {\printfield{title}%
      \clearfield{title}}
    {\printfield{title}{shorttitle}}}
  {\printfield{label}}}

```

从中可以看到, 该宏把 title 域作为标签进行输出, 并在输出后清除了 title 域, 自然, 在完成了整个 author/editor+others/translator+others 宏后, title 域就不存在了, 因此在后面的 publisher+location+date 宏中使用 title 域来进行中英文判断是必然会出错的, 找到了问题所在, 那么就有相应的方法, 比如在动态数据修改时, 把 title 信息存储到 userd 域中, 然后在 publisher+location+date 宏中利用该域的信息进行判断, 就不会出现问题了。

从这样一个查找问题的例子, 我们看到文献著录内容组织的基本原理, 也克服了对 biblatex 复杂代码的恐惧, 接下来就可以按部就班的解决想要解决的问题了。

### 3.1.3 兼容性问题的解决思路

**3.1.3.1 biblatex 版本的兼容性** 由于 biblatex 始终在更新, 因此一些新版本的改动可能导致原来可用的样式失效, 因此需要根据 biblatex 的版本做一定的调整。而要做调整需要知道当前的 biblatex 版本, 以及当前版本做了哪些改动, 是否影响原来的样式文件。

因此样式首先根据 biblatex 版本信息进行版本判断, 然后做针对性的兼容处理。biblatex 在 texlive 中的版本对应关系是: texlive2015 对应 biblatex3.0; texlive2016 对应 biblatex3.4, 从 3.2 版开始修改了姓名机制; texlive2017 对应 biblatex3.7; texlive2018 对应 biblatex3.11;

因为 biblatex 版本升级需要做的一些兼容性处理包括:

宏包选项:

```
firstinits=true,%for texlive2014,2015;bibtex <=3.2
giveninits=true,%for texlive2016,2017;bibtex >=3.3
labeldate=true,%for texlive2014,2015,2016,2017;bibtex <3.8
labeldateparts=true,%for bibtex >=3.8a
```

命令:

```
\printdateextralabel,%for texlive2014,2015,2016;bibtex <=3.4
\printlabeldateextra,%for texlive2017;bibtex >=3.7
\DeclareSortingScheme %for texlive2014,2015,2016,2017;bibtex <3.8
\DeclareSortingTemplate %for bibtex >=3.8a
```

biblatex 的版本判断最初使用一个简单的字符流处理来解析版本信息`\abx@version`。

```
\providetoggle{iftexlivesix}
%\def\versionstr{3.4}
%\def\versionstra{3.6}
%\ifx\abx@version\versionstr
%\toggletrue{iftexlivesix}
%\else
%\ifx\abx@version\versionstra
%\toggletrue{iftexlivesix}
%\else
%\togglefalse{iftexlivesix}
%\fi
%\fi
%改变版本判断机制,根据biblatex更新历史可知,版本3.3开始使用新的姓名处理机制
%所以当版本大于3.2开始,就用设置\toggletrue{iftexlivesix}
\def\numparserta#1.#2\relax{#1}%注意relax的重要性
\def\numparsertb#1.#2\relax{#2}
\def\numinteger{\expandafter\numparserta\abx@version\relax}
\def\numdigital{\expandafter\numparsertb\abx@version\relax}
\ifnumcomp{\numdigital}{>}{2}{\toggletrue{iftexlivesix}}{\togglefalse{iftexlivesix}}
```

但这种方式容易出现问题,于是换成利用 xstring 宏包的命令来提取版本中的数字进行判断:

```
%版本判断
\providetoggle{iftexlivesix}%用于处理biblatex3.3开始改变的新的姓名机制
\providetoggle{iftexliveseven}%用于处理biblatex3.7的兼容性
\providetoggle{iftexliveeight}%用于处理biblatex3.8a的兼容性
%\def\numparserta#1.#2a\relax{#1}%注意relax的重要性
%\def\numparsertb#1.#2a\relax{#2}
%\def\numinteger{\expandafter\numparserta\abx@version\relax}
%\def\numdigital{\expandafter\numparsertb\abx@version\relax}
\StrChar{\abx@version}{3}[\numdigital]
%根据biblatex更新历史,当版本大于3.2开始,就用设置\toggletrue{iftexlivesix}
\ifnumcomp{\numdigital}{>}{2}{\toggletrue{iftexlivesix}}{\togglefalse{iftexlivesix}}
\ifnumcomp{\numdigital}{>}{6}{\toggletrue{iftexliveseven}}{\togglefalse{iftexliveseven}}
\ifnumcomp{\numdigital}{>}{7}{\toggletrue{iftexliveeight}}{\togglefalse{iftexliveeight}}
```

但这种方式仍然会存在问题,特别是当版本信息中带有字母的时候,比如 3.8a 版本。因此进一步考虑了判断的方式,设计了一种新的处理方式便于以后的扩展。这种方式自然兼容最新的版本,但老的版本的代码处理必须要到位,即因为 biblatex 版本升级增加了新的老版本时,对应该老版本的选择必须要实现,否则可能漏掉这些旧版本的处理。

```

%版本判断, 当版本继续更新时, 增加一个新的toggle用以处理新的旧版, 最新版本永远用iftlatest
\providetoggle{iftlfive}%用于处理biblatex3.2之前的版本
\providetoggle{iftlsix}%用于处理biblatex3.3开始改变的新的姓名机制后版本
\providetoggle{iftlseven}%用于处理biblatex3.7的兼容性
\providetoggle{iftleight}%用于处理biblatex3.8到3.9的兼容性
\providetoggle{iftlatest}%用于最新版biblatex, 目前是biblatex3.10, 20180117
\StrBefore{\abx@version}{.}[\numinteger]%以点区分整数和小数
\StrBehind{\abx@version}{.}[\numdigital]%小数部分可能带字母所以需将其去掉, 见下一行代码
\IfInteger{\numdigital}{\StrGobbleRight{\numdigital}{1}[\numdigital]}
\togglefalse{iftlfive}
\togglefalse{iftlsix}
\togglefalse{iftlseven}
\togglefalse{iftleight}
\toggletrue{iftlatest}
\ifnumcomp{\numinteger}{=}{3}{
  \ifnumcomp{\numdigital}{<}{10}{\ifnumcomp{\numdigital}{>}{7}{\toggletrue{iftleight}\togglefalse{
    iftlatest}}}{}}>3.8 版本用iftleight=true表示
  \ifnumcomp{\numdigital}{=}{7}{\toggletrue{iftlseven}\togglefalse{iftlatest}}}{}}3.7 版本用iftlseven=true
  表示
  \ifnumcomp{\numdigital}{<}{7}{\ifnumcomp{\numdigital}{>}{2}{\toggletrue{iftlsix}\togglefalse{iftlatest
    }}}}{}}3.3-3.6 版本用iftlsix=true表示
  \ifnumcomp{\numdigital}{<}{3}{\toggletrue{iftlfive}\togglefalse{iftlatest}}}{}}3.0-3.2 版本用iftlfive=
  true表示
}{\ifnumcomp{\numinteger}{>}{3}{
  \blx@warning@noline{%
    biblatex version is >= 4.x.\MessageBreak
    if errors raised, Please contact biblatex-gb7714-2015 pkg author.}%
}{\toggletrue{iftlfive}\togglefalse{iftlatest}}}%2.x版本统一用iftlfive=true表示
  \blx@warning@noline{%
    biblatex version is <= 2.x.\MessageBreak
    if errors raised, Please contact biblatex-gb7714-2015 pkg author.}%
}}
\newcommand\defversion[2]{\csdef{codeversion#1#2}}%定义不同版本的命令
\newcommand\switchversion[2]{\csuse{codeversion#1#2}}%使用不同版本的命令

\defversion{3.0}{opt}{
  %biblatex3.3版前(比如texlive2015中的3.0版)的使用方式

  %增加标签对齐选项, right是默认的右对齐, left是左对齐, gb7714-2015无效, 仍然为右对齐模式, 因为舍弃了list类环
  境后出错。
  \DeclareBibliographyOption{gbalign}[right]{%texlive2015中的3.0版中的DeclareBibliographyOption选项没有类
  型说明
    \ifstrequal{##1}{right}{}{}
    \ifstrequal{##1}{left}{\setalignleft}{}
    \ifstrequal{##1}{gb7714-2015}{}{}
  }

  \ExecuteBibliographyOptions{
    firstinits=true, %名字有缩写, 参考3.1.2.3 Internal
  }
}

\defversion{3.4}{opt}{
  % Alter settings that carry through from biblatex
  %biblatex3.3版后(比如texlive2016中的3.4版)的使用方式

  %增加标签对齐选项, right是默认的右对齐, left是左对齐, gb7714-2015是项对齐方式
  \DeclareBibliographyOption[string]{gbalign}[right]{%

```

```

\ifstrequal{##1}{left}{\setalignleft}{}
\ifstrequal{##1}{gb7714-2015}{\setaligngbstyle}{}
}

\ExecuteBibliographyOptions{
  giveninits=true,
}

}

\defversion{3.7}{opt}{
  %biblatex3.7版后(比如texlive2017中的3.7版)的使用方式

  %增加标签对齐选项, right是默认的右对齐, left是左对齐, gb7714-2015是项对齐方式
  \DeclareBibliographyOption[string]{gbalign}[right]{%
    \ifstrequal{##1}{left}{\setalignleft}{}
    \ifstrequal{##1}{gb7714-2015}{\setaligngbstyle}{}
  }

  \ExecuteBibliographyOptions{
    giveninits=true,
    date = year, %日期仅写到年
    urldate = edtf, %iso8601
    eventdate = edtf,
  }
}

\defversion{3.10}{opt}{
  %biblatex3.10版后的使用方式

  %增加标签对齐选项, right是默认的右对齐, left是左对齐, gb7714-2015是项对齐方式
  \DeclareBibliographyOption[string]{gbalign}[right]{%
    \ifstrequal{##1}{left}{\setalignleft}{}
    \ifstrequal{##1}{gb7714-2015}{\setaligngbstyle}{}
  }

  \ExecuteBibliographyOptions{
    giveninits=true,
    date = year, %日期仅写到年
    urldate = iso, %iso8601, edtf
    eventdate = iso,
  }
}

\iftoggle{ifftlfive}{\switchversion{3.0}{opt}}{}%biblatex<=3.2
\iftoggle{ifftlsix}{\switchversion{3.4}{opt}}{}%3.3<=biblatex<=3.6
\iftoggle{ifftlseven}{\switchversion{3.7}{opt}}{}%biblatex=3.7
\iftoggle{ifftleight}{\switchversion{3.7}{opt}}{}%3.8<=biblatex<=3.9
\iftoggle{ifftlatest}{\switchversion{3.10}{opt}}{}%biblatex最新3.10

```

版本判断其实也可以考虑其它方法, 比如用宏包的判断命令`\ifpackagelater`, 具体可以参考 nature 样式的做法:

```

\ifpackagelater{biblatex}{2016/03/27}
{\ExecuteBibliographyOptions{giveninits}}
{\ExecuteBibliographyOptions{firstinits}}

% Modify the name format
\ifpackagelater{biblatex}{2016/03/27}
{

```

```

\DeclareNameFormat{default}{%
  \nameparts{#1}%
  \usebibmacro{name:family-given}
    {\namepartfamily}
    {\namepartgiveni}
    {\namepartprefix}
    {\namepartsuffix}%
  \usebibmacro{name:andothers}%
}
}
{
\DeclareNameFormat{default}{%
  \usebibmacro{name:last-first}{#1}{#4}{#5}{#7}%
  \usebibmacro{name:andothers}%
}
}
}

```

还有可能用到一些文档环境的判断方法, 可以使用 `\@ifclassloaded` 命令, 类似的还有判断宏包是否加载的命令, `\@ifpackageloaded` 比如:

```

\@ifclassloaded{beamer}{
  \DeclareFieldFormat{title}{#1}%
  \DeclareFieldFormat[article,patent,thesis,unpublished]{title}{#1}
  \AtBeginDocument{%
    \patchcmd{\abx@macro@title}{\newunitpunct}{-}{-}}{}
}

```

同时也要注意, 可能会用到要在编译不同时刻展开或替换, 代码放置的位置, 比如在宏包末尾, 在导言区末尾, 在正文开始处等:

```

\AtEndOfPackage{code}
\AtEndPreamble{code}
\AtBeginDocument{code}

```

**3.1.3.2 版本兼容的两个途径** 从实践看, 实现不同 biblatex 版本的兼容有两个比较合适的途径。

一是不同版本不同的实现。这是前述选项设置不同版本处理的方法, 不同的版本的选项不一样, 所以针对具体版本设置相应的选项。其他的宏和命令, 也可以类似, 在不同的版本下, 同一个命令有不同的定义, 不同版本使用不同的代码。

二是使用最新的实现代码。这与第一种方法不同, 即不同的版本使用相同的定义, 使用一个代码, 比如最新的代码。但这时可能会出现问题, 因为随着 biblatex 的更新, 最新的代码中可能会使用一些 biblatex 中新增的定义或功能, 而老版本的 biblatex 不具备, 其解决方案是判断当前的版本, 如果是 biblatex 新增的功能不存在就把它从 biblatex 新版中拷贝进样式文件中来补全功能。

两种途径, 一般第一种途径使用较多。第二种也有, 但较少。目前应用的主要是在修改 `textcite` 命令的标点时候用, 由于要在文献表中和标注标签中实现作者域不同的标点, 因此需要不同的文境下面使用不同的定义, 这在较新的版本中通过 `DeclareDelimFormat` 定义不同文境下的标点是很容易实现的。因此考虑使用这种方式, 但由于 `texlive2015` 中的 biblatex 3.0 不具备该功能, 因此对其做补充。

比如 `name:andothers` 宏中使用的 `\printdelim{andothersdelim}`, 在不同的文境下需要不同的定义, 标注中需要是空格, 而文献表中需要是逗号。



```

\DeclareDelimFormat{andothersdelim}{\addcomma\addspace}
\DeclareDelimFormat[textcite]{andothersdelim}{\addspace}
%
% 修改作者数量超过限定值, 做省略时的处理格式
%
% 原理方法: 判断作者或译者是否中文, 若中文用字符等, 否则用etc1.
\renewbibmacro*{name:andothers}{%
  \ifboolexpr{%
    test {\ifnumequal{\value{listcount}}{\value{liststop}}}
    and
    test \ifmorenames
  }%
  {%这里做一个判断是在处理author还是translator用于两者是不同语言的情况
  \ifcurrentname{translator}{\testCJKfirst{usere}}{\testCJKfirst{userf}}}%
  %这句判断如果放到\andothersdelim后面会在等或etc.前增加一个空格, 所以放前面
  \ifnumgreater{\value{liststop}}{1}%
    {\finalandcomma}%
    {%biblatex作者要区别单作者加等的情况, 这里为符合gbt7714-2015第7.2节的要求加上了逗号。
  \printdelim{andothersdelim}\iftoggle{ifCJKforgbt}{\bibstring{andotherscn}}{\bibstring{andothers}}}%
  %\andothersdelim\bibstring{andotherscn}
}%
{}}

```

但 biblatex3.0 没有相应功能, 为简便起见, 直接将 DelimFormat 相关的功能补充进来。当然这仍然可以采用前述的第一种方法来解决, 针对 biblatex3.0 做针对性的实现, 即不考虑不同文境下的标点输出, 而直接在标注命令中根据版本进行标点的局部重定义。两中途径主要根据方便与否进行选择, 这里的问题选择第二种途径, 因此补充 DelimFormat 相关功能:

```

\iftoggle{ifftlfive}{

\def\blx@inf@delimdeclare#1#2{%
\blx@info{Delimiter '#1' in context '#2' already defined, overwriting}}

\def\blx@warn@delimuse#1#2{%
\blx@warning{Delimiter '#1' in context '#2' undefined}}
% Delimiter interface

% [<contextname, ...>]{<name, ...>}{<code>}
\newrobustcmd*{\DeclareDelimFormat}{%
  \@ifstar
  {\blx@declaredelimclear}
  {\blx@declaredelim}}

\newrobustcmd*{\blx@declaredelimclear}[3][]{%
  \ifcsvoid{blx@declaredelimcontexts@#2}
  {}
  {\def\do##1{\csundef{blx@printdelim@##1@#2}}%
   \dolistcsloop{blx@declaredelimcontexts@#2}%
   \cslet{blx@declaredelimcontexts@#2}\@empty
   \ifblank{#1}
   {\blx@declaredelim{#2}{#3}}
   {\blx@declaredelim[#1]{#2}{#3}}}

\newrobustcmd*{\blx@declaredelim}[3][]{%
  \ifblank{#1}
  {\blx@declaredelim@i{#2}{#3}}
  {\def\do##1{%
    \listcsadd{blx@declaredelimcontexts@#2}{##1}%

```

```

\blx@declaredelim@i{blx@printdelim@##1@}{##1}{#2}{#3}}%
\docsvlist{#1}}}%

\def\blx@declaredelim@i#1#2#3#4{%
\def\do@i##1{%
\ifcsdef{#1##1}
{\blx@inf@delimdeclare{##1}{#2}}
{}}%
\csdef{#1##1}{#4}}%
\forcsvlist{\do@i}{#3}}

% * [<contextname, ...>]{<alias>}{<delim>}
\newrobustcmd*{\DeclareDelimAlias}{%
\ifstar
{\blx@declaredelimalias}
{\blx@declaredelimaliasauto}}

\newrobustcmd*{\blx@declaredelimalias}[3][{}]{%
\ifblank{#1}
{\blx@declaredelimalias@i{#2}{#3}}
{\def\do##1{%
\blx@declaredelimalias@i{blx@printdelim@##1@}{#2}{#3}}%
\docsvlist{#1}}}}

\newrobustcmd*{\blx@declaredelimaliasauto}[2]{%
\blx@declaredelimalias@i{#1}{#2}%
\ifcsvoid{blx@declaredelimcontexts@#2}
{}
{\def\do##1{%
\blx@declaredelimalias@i{blx@printdelim@##1@}{#1}{#2}}%
\dolistcsloop{blx@declaredelimcontexts@#2}}}}

\def\blx@declaredelimalias@i#1#2#3{%
\ifcsdef{#1#2}
{\blx@inf@delimdeclare{#2}{#1}}
{}}%
\csdef{#1#2}{\csuse{#1#3}}}}

\def\blx@delimcontext{none}
\newcommand*{\printdelim}[2][{}]{%
\ifblank{#1}
{\ifcsdef{blx@printdelim@blx@delimcontext @#2}
{\csuse{blx@printdelim@blx@delimcontext @#2}}
{\ifcsdef{#2}% fall back on legacy macros
{\csuse{#2}}
{\blx@warn@delimuse{#2}{*}}}}
{\ifcsdef{blx@printdelim@#1@#2}
{\csuse{blx@printdelim@#1@#2}}
{\blx@warn@delimuse{#2}{#1}}}}

\newcommand*{\delimcontext}[1]{%
\edef\blx@delimcontext{\blx@delimcontextalias{#1}}}

\def\blx@delimcontextalias#1{%
\ifcsdef{blx@delimcontextalias@#1}
{\csuse{blx@delimcontextalias@#1}}
{#1}}

```

```
\newcommand*{\DeclareDelimcontextAlias}[2]{%
  \csgdef{blx@delimcontextalias@#1}{#2}}
}{}
```

**3.1.3.3 GBK 编码文件的兼容性** 在使用纯英文的文档中,不涉及到文件的编码问题。但在中文文档中可能涉及到文件编码的问题,因为默认情况下 windows 下创建的文件是 cp 936 编码即 GBK 编码,如果不将其转换为 utf-8 编码,那么就会涉及到该问题。

我们知道 gbk 编码的 tex 文档,利用 pdflatex/latex 能正确编译,利用 xelatex 能编译,但中文会显示乱码,在源文档前面增加 XeTeX 原语: `\XeTeXinputencoding "GBK"` 后,显示正常。而 utf-8 编码的 tex 文档,利用 xelatex 能正确编译,而 pdflatex/latex 则不能编译。总的来说,使用 GBK 编码的情况现在已经大为减少,而遇到时通常是使用 pdflatex/latex 编译的。因此在参考文献样式中也将面临这一问题。

在 biblatex 中提供了 texencoding 和 bibencoding 选项,它是通过将输入的选项值传递给 biber,然后 biber 利用 perl 的 encode 模块实现编码转换的。因此只要是 perl 的 encode 模块支持的编码转换,biblatex 都是支持的。perl 的 encode 模块的信息见 [encode](#),中文的编码信息见 [encode::cn](#)。用法可以参见: [perl 文件编码处理问题](#)。

当使用 GBK 编码时,我们可以在 biblatex 中使用选项:

```
texencoding=GBK
bibencoding=GBK
```

然而当 biblatex 的样式文件中含有一些中文字符串时,仍然会出现问题,因为这些字符与样式文件一样是 utf-8 编码的。因此要把这些字符串替换掉。这时可以使用命令来替换字符,然后用 GBK 编码的文件中的字符串来重定义这些命令。这一解决思路来自于 [biblatex-casparvector](#)。而且需要注意的是,当这一替换在样式文件内部进行时,仍然会出现问题,因此还需要在宏包结尾来进行替换。

处理代码如下:

```
%
% 增加一个源文件编码选择选项,当true时可以使用GBK编码。
% v1.0k, 2018.05.08, by hzz
%
% 原理方法:将需要输出的中文字符串用命令形式表示存储,采用utf8编码时采用bbx文
% 档内的定义即是utf8编码的中文字符串,但tex源文件是gbk编码时,利用gb7714-2015-gbk.def
% 文档内的gbk编码的字符串覆盖。这种覆盖只能在宏包末尾加载时成功实现,所以使用
% AtEndOfPackage。该解决思路学自biblatex-casparvector宏包。
% 另外注意:
% gbk编码的tex文档,利用pdflatex/latex能正确编译,利用xelatex能编译,但中文显示乱码
% 在源文档前面增加 XeTeX 原语:\XeTeXinputencoding "GBK" 后,显示正常
% utf-8编码的tex文档,利用xelatex能正确编译,而pdflatex/latex不能编译。
\newtoggle{bbx:codegbk}
\DeclareBibliographyOption{gbcodegbk}[false]{%
  \settoggle{bbx:codegbk}{#1}}
\ExecuteBibliographyOptions{gbcodegbk}

\def\str@bibliography{参考文献}
\def\str@references{参考文献}
\def\str@bytranslator{译}
\def\str@andotherscn{等}
\def\str@backrefpage{引用页}
\def\str@backrefpages{引用页}
```

```

\def\str@noaddress{出版地不详}
\def\str@nopublisher{出版者不详}
\def\str@edition{版}

\AtEndOfPackage{%
  \iftoggle{bbx:codegbk}{%
    \ExecuteBibliographyOptions{texencoding = GBK}
  }
  \input{gb7714-2015-gbk.def}
}

```

测试文件见：[eggbk.tex](#)，[eggbkay.tex](#)。注意编译时请使用 `pdflatex` 或 `latex`。

同时也注意到，`bbx` 文件处理“佚名”和“等”这些本地化字符串之间的差别，因为处理“佚名”的动态数据修改始终是在 `perl` 中进行的，所以直接给出中文字符没有问题。而“等”这些本地化字符串必须要用 `GBK` 编码的文件中的重定义进行覆盖。

另外还要注意的，`GBK` 编码的文件不能使用中文的 `bibtex` 键即中文的引用关键字。

### 3.1.4 中英文混合文献表的解决思路

考虑到我国引用参考文献通常是中英文混合的情况，因此不考虑针对中文的本地化处理，而是在英文本地化的基础（英文的本地化字符串设置文档是 `english.lbx`）上添加一些中文要求的本地化字符串来使用。

需要注意：本地化字符串的定义在 `lbx` 文件和 `bbx` 文件中是不一样的，`lbx` 文件中用的是 `DeclareBibliographyStrings`，而 `bbx` 文件中用的是 `DefineBibliographyStrings`，两者的参数是不一样的。

而为了区分使用中英文的字符串，对参考文献条目中数据域内容进行中英文判断，若是中文则使用中文字符串，若是英文则使用英文字符串。这里的中英文判断主要是在 `tex` 中的判断，利用对域中的信息进行检测，当信息的第一个字符是 `CJK` 字符时，判断该域的信息是中文的，否则是英文的。

在作者年制中不同语种文献分集时也有一个中英文判断，主要是利用动态数据修改的方法。利用 `perl` 正则表达式判断作者和标题域中是否有中文信息（可以用排除法即匹配不是英文数字标点的字符通常是中文字符，当然最直接的是用 `unicode` 的中文字符范围匹配），有的话设置 `cn`，否则为 `en`。

事实上这种判断可以进一步扩展，利用强大的 `perl` 正则表达式，对所有的关键域进行判断，并在一些域中设置标识，用于 `tex` 输出时进行判断，而避免直接使用上述 `CJK` 字符判断函数。

#### 3.1.4.1 利用字符流解析或者 `xstring` 宏包的函数 以实现中英文的判断

`latex` 判断字符属于哪个区间的方法，最直接的思路就是利用`\`获取字符的编码，然后利用编码进行判断。这里 `testCJKfirstchar` 函数就是这样的思路。

字符编码获取的简单方法如下：

```

\newcounter{charcode}
\setcounter{charcode}{`文}
编码为：\thecharcode

```

`cjk` 字符判断函数如下：

```

%定义判断中文字符的函数，用于判断作者等信息是否由中文字符构成
%2E00-2E7F 追加标点

```

```
%2E80-2EFF cjk部首补充
%2FF0-2FFF 表意文字描述符
%3000-303F cjk符号和标点
%3300-33FF cjk兼容
%3400-4DBF cjk统一表意符号扩展
%4E00-9FBF cjk统一表意符号
%中文范围4E00-9FA5
\providetoggle{ifCJKforgbt}
\def\testCJKfirstchar#1#2&{%
\ifnumgreater{`#1}{`"2E7F}{\toggletrue{ifCJKforgbt}}{\togglefalse{ifCJKforgbt}}}%
}%

%利用\edef展开或xstring抽取第一个字符判断
%现在采用xstring方法, 避免抽取的是编组符号
\def\testCJKfirst#1{%
%\edef\tempa{#1}%可以替换下面两句, 但失去忽略多余编组功能
\exploregroups%
\StrChar{#1}{1}[\tempa]%
\expandafter\testCJKfirstchar\tempa&}
```

测试文件见: [testcharcode-utf8-xelatex.tex](#), [testcharcode-gbk-xelatex.tex](#), [egtestcjk-utf.tex](#)。

然而利用```获取编码的方法只对原始支持 unicode 编码的引擎有效, 比如 xelatex。当使用 GBK 编码时, 如果加 xetex 原语<sup>9</sup>后使用 xelatex 编译也能够获得字符编码, 但当使用 pdflatex 或 latex 编译时, 就不能获取, 因为```的意义变了。因此继续使用上述方法就会出现問題, 因此考虑刘海洋提供的利用 CJKsymbol 进行判断的思路<sup>10</sup>, 方法如下:

### 3.1.4.2 利用 CJKsymbol 进行的 cjk 判断 以实现中英文的判断

函数如下:

```
%
% 判断CJK字符的函数, 用于判断作者等信息是否由中文字符构成
% v1.0k, 20180509, hzz
%
% 原理方法: 因为考虑使用gbk编码的问题, 将原来基于`加字符获取字符编码的cjk判断
% 函数更换成为利用\CJKsymbol的方法。解决方案由刘海洋提供。因为在gbk编码的文件
% 中使用pdflatex编译时`不再是xelatex中的意义。当然这里利用tex的判断方法也可以
% 用biber中的perl方法代替。当前使用这一函数能解决问题情况下, 暂不做修改。
%
% 原理是: CJK 输出汉字最终会使用 \CJKsymbol。
% 因此重定义 \CJKsymbol, 以在这个宏中设置\ifCJK为真。
% 注意这时整个字符串内只要有cjk字符那么就会判断为cjk。
\providetoggle{ifCJKforgbt}
\def\testCJKfirst#1{%
\begingroup%
\global\togglefalse{ifCJKforgbt}%
\def\CJKsymbol##1{\global\toggletrue{ifCJKforgbt}}%
\setbox0=\hbox{#1}%
\endgroup}%

```

测试文件见: [egtestcjk-utf-xelatex.tex](#), [egtestcjk-gbk-pdflatex.tex](#)。

然而, 该方法解决了 gbk 编码的问题, 但又会导致 utf 编码时 xelatex 编译产生问题。当直接对字符进行判断时没有问题, 但当字符时定义在一个宏中时, 就会出现缺少`}`的问题。比如:

<sup>9</sup><http://blog.chinaunix.net/uid-23755487-id-2976433.html>

<sup>10</sup><http://bbs.ctex.org/forum.php?mod=viewthread&tid=152663>

```

\documentclass[UTF8]{ctexart}
\usepackage{etoolbox}
\usepackage{xstring}

\newif\ifCJK
\def\testCJK#1{%
  \begingroup
  \global\CJKfalse
  \def\CJKsymbol##1{\global\CJKtrue}
  \setbox0=\hbox{#1}%
  \endgroup}

\def\hztmp{汉}

\edef\hztmpb{汉}

\begin{document}

\testCJK{Eng}
\ifCJK CJK\else Eng\fi

\testCJK{eng汉字}
\ifCJK CJK\else Eng\fi

%\testCJK{\hztmp}
%\ifCJK CJK\else Eng\fi
%
%\testCJK{\hztmpb}
%\ifCJK CJK\else Eng\fi

%\expandafter\testCJK\hztmp
%\ifCJK CJK\else Eng\fi
%
%\expandafter\testCJK\hztmpb
%\ifCJK CJK\else Eng\fi

\testCJK\hztmp
\ifCJK CJK\else Eng\fi

\testCJK\hztmpb
\ifCJK CJK\else Eng\fi
\end{document}

```

因此，只能利用最终的办法，就是利用动态数据修改中 perl 进行直接标识。

**3.1.4.3 中文的本地化字符串** 在英文的本地化文件基础上增加修改定义的字符串，以满足中文的规范，比如：

```

\DefineBibliographyStrings{english}{
  bibliography = {参考文献},
  references = {参考文献},
  %bytranslator= {\addcomma\ 译\addperiod}, %将trans. by 改成 译
  bytranslator= {\addcomma\ 译},%\addperiod
  and = {\addcomma},%将第2和3人名见的and符号改成 逗号，用\finalnamedelim命令也可以定义，参见3.9.1节
  %andothers = {et al.}, %将超过3个人名的省略，et al.改成为 等
  andotherscn = {等}, %将超过3个人名的省略，et al.改成为 等
  noaddress = {[S.l.]},
  nopublisher = {[s.n.]}
}

```



}

### 3.1.5 功能增加及格式还原的解决思路

GB 样式是在标准样式基础上修改的, 因此只要还原每一个修改, 那么最终一定能够从 gb 样式还原到标准样式。为了方便控制这种修改和还原, 采用宏包选项控制的方式进行, 因此需要增加相应的控制选项。

从总体功能上看, GB 样式与标准样式的区别主要有:

- 中英文判断问题及对应的缩略词输出问题
- GB 样式中对于期刊有起止卷期的要求
- GB 样式对于多语言文献对照的要
- GB 样式对于作者缺省时处理的要求
- GB 样式对于顺序编码制文献表标签对齐要求

其中前三项在本节其它地方介绍, 后面两项可以通过选项来进行控制。作者缺省时的处理用选项 `gbnoauthor` 控制还原, 文献表标签对齐用 `gbalign` 选项控制还原。

从著录格式上看, GB 样式与标准样式的区别主要有:

- 作者域格式
- 标题、网址、页码等其他域的格式
- 出版项缺省的格式
- 标点
- 文献类型和载体标识
- 对一些域的输出做了限制

作者域格式利用选项 `gbnamefmt` 还原。标题等域的格式利用选项 `gbfieldstd` 还原。出版项缺省的格式利用选项 `gbpub` 还原。标点比如析出源之前的//利用选项 `gbpunctin` 还原。其它域比如卷等, 在标准样式中是区分不同的文献条目类型, 比如书籍或期刊文章中卷格式是不同的, 书籍中加了 `vol` 引导词, 而期刊文章中没有。而 gb 样式中 [年份, 卷 (期)] 这种方法是比较清晰的格式, 在一些样式如 APA 等也有应用, 其实也可不考虑还原回, 甚至也可以使用另一种常见的全部加引导词的方式, 类似 iso690 样式的做法。而对于域出现的顺序, 则不做还原, 避免大量的代码改动。对于原来标准样式中输出的域做的限制, 也可以利用选项取消, 这里采用 `gbstrict` 选项还原。

总的来说, 尽管 GB/T 7714 样式看起来并不一定让人觉得美观, 但我认为它是经过深入设计的, 特别是考虑了中英文文献混合的情况, 当采用某些英文文献常见做法 (比如使用引导词) 时, 这种英文的引导词其实并不适合于中文, 而中文又没有对应的合适的引导词, 所以还不如用标点来代替。比如用//代替 `in`, 用 [年份, 卷 (期)] 代替 [年份, `vol.` 卷, `no.` 期], 用: 代替 `pp.` 等。但域内容的一些格式、字体等是可以根据需要进行调整的, 以达到更美观的效果。

**3.1.5.1 biblatex 的宏包选项增加方法** 要利用 `biblatex` 的选项机制来实现某些目的, 需要首先弄清楚 `biblatex` 及其样式文件中的一些判断/选择机制。

在 `gb7714-2015.bbx` 文件中有四种常见的判断方式包括: 版本的判断, 宏包选项的判断, 宏包选项控制的切换标记的判断, 一般的域是否定义或与字符串比较是否相同的判断。

第一种判断, 是根据已有的信息直接判断, 比如利用 `\abx@version` 或 `\abx@date` 进行判断, 这种处理结果是直接展开的, 可以用于后面的任意定义。

第二种判断, 是根据不执行默认设置的选项进行判断。例如 (注意该选项是 `string` 类型):

```
\DeclareBibliographyOption[string]{gbalign}[right]{%
  \ifstrequal{##1}{left}{\setalignleft}{}
  \ifstrequal{##1}{gb7714-2015}{\setaligngbstyle}{}
}
```

这里定义的选项 `gbalign`，但不利用 `ExecuteBibliographyOptions` 命令执行默认的设置。这时宏包会按照没有该选项影响的方式进行处理。只有当在 `biblatex` 加载时，该选项才会有意义。

当选项给出时，宏包会执行选项中对应选项值的代码，比如给出为 `gbalign=left`，则执行 `setalignleft` 命令。

还要注意：当加载选项又不设置选项值时，宏包会执行 `DeclareBibliographyOption` 命令中给出的默认选项，也就是说，当给出选项 `gbalign`，等价于给出 `gbalign=right`。但此时选项定义中没有对应该 `right` 值需要再执行的命令，既不执行 `setalignleft`，也不执行 `setaligngbstyle`，因此不做任何处理。只有设置宏包选项为 `left` 或 `gb7714-2015` 时才会调用 `setalignleft` 和 `setaligngbstyle` 命令做进一步的处理。

第三种判断，根据宏包选项控制对切换标记进行设置，然后在代码中应用切换标记判断。比如 `standard.bbx` 中定义的 `url` 等选项。

```
\newtoggle{bbx:gbpub}
\DeclareBibliographyOption[boolean]{gbpub}[true]{%
  %\settoggle{bbx:gbpub}{#1} %或采用下面这一句
  \ifstrequal{##1}{false}{\togglefalse{bbx:gbpub}}{\toggletrue{bbx:gbpub}}
}\ExecuteBibliographyOptions{gbpub}
```

注意到在上述选项定义后面，使用了选项默认执行命令，这时宏包选项定义中的代码会立刻被执行，即在样式中设置了 `gbpub=true`。

注意，样式中可以利用切换标记 `bbx:gbpub` 进行任意的宏定义。但如果该宏展开层次是与这里定义的宏包选项同一层次的，那么切换标记会根据默认值是 `false` 而展开，因为宏该宏展开时，选项的执行还未进行。如果宏是下一层的展开（即不需要在加载时展开的），那么会根据展开时，切换标记 `bbx:gbpub` 的值进行展开。即这种机制的切换标识判设置，能用于 `usemacro` 使用或者域格式定义中，但无法用来判断然后再定义 `macro`，因为定义宏和使用宏是两个不同的展开层级。

注意 `ExecuteBibliographyOptions` 命令是用来执行选项的，如果不给出这一命令，且宏包加载选择中没有给出选项设置（给出选项等价于执行 `ExecuteBibliographyOptions`），那么 `toggle` 只有新建 `toggle` 时的默认定义 `false`。

还要注意该命令执行选项是立刻进行的，如果选项的定义代码不是设置切换标记，而是直接调用命令，比如前面的 `gbalign` 的情况，那么就需要被调用命令是有定义的，即命令的定义至少与宏包选项定义在同一展开层次，甚至先于宏包选项定义。

还需要注意，如果给出了命令

```
\ExecuteBibliographyOptions{gbpub}%等价于下一条命令
\ExecuteBibliographyOptions{gbpub=true}
\ExecuteBibliographyOptions{gbpub=false}
```

等价于在宏包加载时给出选项。类似的，如果在宏包加载时给出选项 `gbpub=false`，那么 `bbx:gbpub` 的值设置为 `false`。如果在宏包加载时给出选项 `gbpub`，`gbpub=true`，那么 `bbx:gbpub` 的值设置为 `true`。但要注意宏包加载选项设置可以覆盖这个命令的设置，因为通常随宏包加载选项会把选项放在样式选项之后。而后面的选项设置会覆盖前面的设置。

第四种判断，利用 `iffielddundef` 和 `iffieldequalstr` 进行判断即可。域内容的判断方法比如：

```
%方法1
\iffieldequalstr{note}{news}{\usebibmacro{newsdate}}}%判断是否为报纸
{\usebibmacro{date}}}%

%方法2
\def\comparatmp{news}\edef\comparatmpb{\thefield{note}}%
\ifx\comparatmp\comparatmpb%
\usebibmacro{newsdate}%
\else%
\usebibmacro{date}%
\fi
```

总结起来, 宏包选项的问题需要注意宏包选项的展开顺序。宏包选项的定义如果不使用 `ExecuteBibliographyOptions` 命令或者不在加载时设置, 那么定义的代码是不执行的。因此如果其他代码中需要使用该宏包选项定义代码中的信息比如切换标记是无法正确获取的。而当使用 `ExecuteBibliographyOptions` 命令或者加载选项时, 那么宏包选项代码会立即展开, 其中需要使用的宏则需要有定义, 即先于或与宏包选项定义处于同一展开层。

### 3.1.5.2 作者缺省处理 利用选项 `gbnoauthor` 控制

```
%增加一个处理佚名或noauthor的控制选项
\DeclareBibliographyOption{gbnoauthor}[true]{%
  \ifstrequal{##1}{false}{\deallnoathor}}
%\ExecuteBibliographyOptions{gbnoauthor=true}
```

这一选项的机制是, 默认情况下不执行选项, 这时按照 `bbx` 中给出的默认方式进行处理。当加载 `biblatex` 时, 给出选项, 如果是 `false`, 则不做任何处理。如果是 `true`, 则调用 `deallnoathor` 命令, 执行内部的代码, 以产生‘佚名’等信息来补充缺失的作者。

`deallnoathor` 的处理代码:

```
%
% 佚名/noauthor的处理
%
% 原理方法:利用样式层的动态数据修改实现
\def\deallnoathor{
  \DeclareStyleSourceMap{
    \maps[datatype=bibtex]{
      \map{%默认情况下将空缺作者设置为佚名
        \step[fieldsource=userb,match={cn},final]
        \step[fieldset=author, fieldvalue={佚名}}
      }
      \map{%默认情况下将空缺作者设置为佚名
        \step[fieldsource=userb,match={en},final]
        \step[fieldset=author, fieldvalue={NOAUTHOR}]
      }
    }
  }
}
```

### 3.1.5.3 文献表标签对齐 利用选项 `gbalgn` 控制

```
%
% 增加标签对齐选项
%
% right是默认的右对齐, left是左对齐, gb7714-2015是项对齐方式
\DeclareBibliographyOption[string]{gbalgn}[right]{%
```

```

\ifstrequal{##1}{left}{\setalignleft}{}
\ifstrequal{##1}{gb7714-2015}{\setaligngbstyle}{}
}

```

这一选项的机制是，默认情况下不执行选项，这是按照 bbx 中给出的默认方式即右对齐处理标签。当加载 biblatex 时，给出选项，如果是 left，则调用 setalignleft 命令，利用其中的代码产生左对齐效果。如果是 gb7714-2015，则调用 setaligngbstyle 命令，执行内部的代码，以实现类似 word 中的项对齐效果。

```

%
% 顺序编码制标签对齐方式处理
%
% 原理方法:修改序号标签格式为左对齐,注意各参考文献内容还是对齐的,
% 这样就会使得序号标签与参考文献内容的间隔增大,这个问题是没有办法解决的
% 因为采用list做具有一定宽度的序号标签, \labelwidth只能设置一个,且是最宽的标签的宽度
% 但总的来说参考文献内容对齐是合理和漂亮的,
% 而标签则只能对齐一个方向,要么左对齐要么右对齐,看个人选择了。
% \DeclareFieldFormat{shorthandwidth}{\mkbibbrackets{#1}} %来自numeric.BBX
% \DeclareFieldFormat{labelnumberwidth}{\ttfamily\mkbibbrackets{#1}\hfill}
% \iftoggle{alignleft}{\DeclareFieldFormat{labelnumberwidth}{\mkbibbrackets{#1}\hfill}}{}% 这种方式无效是因
% 为加载时就已经展开了。
%
% 修改序号标签格式为左对齐
\def\setalignleft{\DeclareFieldFormat{labelnumberwidth}{\mkbibbrackets{##1}\hfill}}
%
% 修改序号标签格式为以各条参考文献为基础进行对齐的方式,即序号与条目内容间隔相等的方式。
\def\setaligngbstyle{%
\def\blx@bibitem##1{%
\blx@ifdata{##1}
{
\begingroup
\blx@getdata{##1}%
\blx@bibcheck
\iftoggle{blx@skipentry}{}{%
\blx@setdefaultrefcontext{##1}%
\global\let\blx@noitem\empty
\blx@setoptions@type\abx@field@entrytype
\blx@setoptions@entry
\blx@thelabelnumber
\addtocounter{instcount}\@ne
\blx@initsep
\blx@namesep
\csuse{blx@item@\blx@theenv}\relax
% \blx@initsep %移动到上面去,恢复bibnamesep等的作用机制
% \blx@namesep
\csuse{blx@hook@bibitem}%
\blx@execute
\blx@initunit
\blx@anchor
\blx@beglangbib
\blx@sentence
\blx@pagetracker
\blx@driver\abx@field@entrytype
\blx@postpunct
\blx@endlangbib}%
\par\endgroup}%这里增加了一个\par
}}
\newlength{\lengthid}
\newlength{\lengthlw}

```

```

\newcommand{\itemcmd}{%
\settowidth{\lengthid}{[\printfield{labelnumber}]}
\addtolength{\lengthid}{\biblabelsep}
\setlength{\lengthlw}{\textwidth}
\addtolength{\lengthlw}{-\lengthid}
\addvspace{\bibitemsep}%恢复\bibitemsep的作用
%\parshape 2 0em \textwidth \lengthid \lengthlw
\hangindent\lengthid
[\printfield{labelnumber}]\hspace{\biblabelsep}}
\defbibenvironment{bibliography}
{\begingroup\setlength{\parindent}{0em}}
{\endgroup}
{\itemcmd}}

```

### 3.1.5.4 作者域格式 利用选项 gbnamefmt 控制

```

%
% 增加姓名大小写格式处理选项
%
\DeclareBibliographyOption[string]{gbnamefmt}[uppercase]{%
  %\ifstrequal{##1}{gbuppercase}{\}{}%
  \ifstrequal{##1}{lowercase}{\execgblowercase}{}%
  \ifstrequal{##1}{none}{\execnamedefault}{%
  \ifstrequal{##1}{pinyin}{\execnamepinyin}{%
  \ifstrequal{##1}{apa}{\execnameapa}{%
}
%\ExecuteBibliographyOptions{gbnamefmt}

```

这一选项的机制是,首先在 `bbx` 中定义了姓名域格式比如 `name:given-family` 等宏,然后根据输入的选项执行 `execgblowercase` 等命令,其中有重定义的姓名域格式。当宏包加载时给出选项设置,那么就会完成姓名域格式重定义。

作者的数量由 `biblatex` 选项控制,作者年制的参考文献表和标签中的作者名数量做区分。列表中最大为 3 个,引用中最大为 1 个。

```

\ExecuteBibliographyOptions{
  %maxnames = 3 , %设置名字最大数量
  %minnames = 3, %设置缩减后的名字最小数量
  maxbibnames=3, %将文献列表和引用中最大名字数量区分开
  minbibnames=3,
  maxcitenames=1,
  mincitenames=1
}

```

### 3.1.5.5 标题等的域格式 利用选项 gbfieldstd 控制

```

%
% 增加一个恢复域格式为标准样式的控制选项
% v1.0k, 2018.05.15, by hzz
%
% 原理方法:
% 默认设置该选项为false, 采用国标要求的域格式输出相关域
% 当设置该选项为true时, 则重新利用标准样式的域格式输出相关的域
\newtoggle{bbx:gbfieldstd}
\DeclareBibliographyOption{gbfieldstd}[false]{%
  \settoggle{bbx:gbfieldstd}{#1}%
  \ifstrequal{#1}{false}{\}{}{\execgbfdmstd}%
}

```

```
}
\ExecuteBibliographyOptions{gbfieldstd}
```

这一选项里面有两个机制，默认情况下执行选项为 false，此时设置 toggle: bbx:gbfieldstd 为 false，不执行其它命令，此时采用 bbx 文件中默认定义的域格式输出，同时在某些宏中根据 bbx:gbfieldstd 标记为 false 输出相应的代码。当设置选项为 true 时，执行 execgbdfmtstd 命令中的代码重定义对应的域格式，并在某些宏中输出对应 toggle 为 true 的代码。

出版项的处理代码为 (注意其中的标点处理):

```
\renewbibmacro*{publisher+location+date}{\bibpubfont%
\iftoggle{bbx:gbpub}%
{\testCJKfirst{userd}%
\iflistundef{location}{%\adddot
\iffieldequalstr{note}{standard}}{从gbt7714-2015标准第19页看到，标准存在出版项时输出，没有时完全省略。
\iftoggle{ifCJKforgbt}{\printtext{\str@noaddress}\addcolon\addspace}{\printtext{[S.1.\adddot]}\addcolon\addspace}%
}}% \bibstring{noaddress}
{\printlist{location}\addcolon\addspace}%
%\addcolon\addspace%
\iflistundef{publisher}{%
\iffieldequalstr{note}{standard}}{%
\iftoggle{ifCJKforgbt}{\printtext{\str@nopublisher}\setunit{\adddot\addspace}\setunit*{\addcomma\addspace}}%
{\printtext{\mkbibbrackets{s.n.}}\setunit{\adddot\addspace}\setunit*{\addcomma\addspace}}%
}}%
{\printlist{publisher}\setunit*{\addcomma\addspace}}%
%\addcomma\addspace%
\usebibmacro{date}%
%\newunit %去掉这个标点
}%
{\printlist{location}%
\iflistundef{publisher}
{\setunit*{\addcomma\space}}
{\setunit*{\addcolon\space}}%
\printlist{publisher}%
\setunit*{\addcomma\space}%
\usebibmacro{date}%
%\newunit
}%
}
```

### 3.1.5.6 出版项缺省的格式 利用选项 gbpub 控制。

```
%
% 增加一个出版项自动处理控制选项，当true时使用出版者不详等信息补充缺失的出版信息。
%
\newtoggle{bbx:gbpub}
\DeclareBibliographyOption{gbpub}[true]{%
%\settoggle{bbx:gbpub}{#1} %或采用下面这一句
\ifstrequal{#1}{false}{\togglefalse{bbx:gbpub}}{\toggletrue{bbx:gbpub}}
\ExecuteBibliographyOptions{gbpub}
```

这个选项的机制是默认执行选项为 true，进而设置 bbx:gbpub 这一 toggle 为 true。bbx 中的处理出版项的宏根据该 toggle 输出相应的代码。



### 3.1.5.7 标点格式 特别是析出源之前的//利用选项 gbpunctin 控制。

```
%
% 增加控制析出文献来源前的标点符号//输出的选项
% v1.0k, 2018.04.20, added in by hzz
\newtoggle{bbx:gbpunctin}
\DeclareBibliographyOption{gbpunctin}[true]{%
  \settoggle{bbx:gbpunctin}{#1}}
\ExecuteBibliographyOptions{gbpunctin}
```

这个选项的机制是默认执行选项为 true，进而设置 bbx:gbpunctin 这一 toggle 为 true。bbx 中的处理出版项的宏根据该 toggle 输出相应的代码。

### 3.1.5.8 文献类型和载体标识 利用选项 gbtype 控制。

```
%
% 增加一个控制是否输出文献类型和载体标识的选项
%
% 原理方法:
% 对于biblatex3.4以上版本DeclareBibliographyOption命令中的[datatype]如果是boolean, 那么是可以省略的
% 所以用老版本的不用[datatype]的命令可以兼容所有biblatex版本
\newtoggle{bbx:gbtype}
% \DeclareBibliographyOption[boolean]{gbtype}[true]{%biblatex高版本
\DeclareBibliographyOption{gbtype}[true]{%biblatex低版本
  \settoggle{bbx:gbtype}{#1}}
\ExecuteBibliographyOptions{gbtype}
```

这个选项的机制是默认执行选项为 true，进而设置 bbx:gbtype 这一 toggle 为 true。bbx 中的处理出版项的宏根据该 toggle 输出相应的代码。

文献载体的输出用标题域格式进行实现，由于 [] 的存在，因此对于标点的处理特别小心：

```
%
% 新增文献类型标识符的格式
%
\DeclareFieldFormat{gbtypeflag}{%
\iftoggle{bbx:url}{\iffieldundef{url}%当存在url时, 增加一个OL标识符
{\nobreak\printtext{[]\nobreak#1\nobreak\printtext{\iffieldundef{medium}{}\{\texttt{/}\thefield{medium}}}}}%
{\nobreak\printtext{[]\nobreak#1\nobreak\printtext{\texttt{/}OL}}}%
}{\nobreak\printtext{[]\nobreak#1\nobreak\printtext{\iffieldundef{medium}{}\{\texttt{/}\thefield{medium}
}}}}}%

%
% 新增用于报纸的文献类型标识符的格式
%
\DeclareFieldFormat{gbtypeflagn}{%用于报纸newspaper
\iftoggle{bbx:url}{\iffieldundef{url}%当存在url时, 增加一个OL标识符
{\nobreak\printtext{[]\nobreak N\printtext{\iffieldundef{medium}{}\{\texttt{/}\thefield{medium}}}}\nobreak}%
{\nobreak\printtext{[]\nobreak N\printtext{\texttt{/}OL}}\nobreak}%
}{\nobreak\printtext{[]\nobreak N\printtext{\iffieldundef{medium}{}\{\texttt{/}\thefield{medium}}}}\nobreak
}}

%
% 新增用于标准的文献类型标识符的格式
%
\DeclareFieldFormat{gbtypeflags}{%用于标准standard
\iftoggle{bbx:url}{\iffieldundef{url}%当存在url时, 增加一个OL标识符
{\nobreak\printtext{[]\nobreak S\printtext{\iffieldundef{medium}{}\{\texttt{/}\thefield{medium}}}}\nobreak}%
{\nobreak\printtext{[]\nobreak S\printtext{\texttt{/}OL}}\nobreak}%
}
```

```

}\nbreak\printtext{[]\nbreak S\printtext{\iffieldundef{medium}{-}{\texttt{/}\thefield{medium}}}\nbreak
}}

\DeclareFieldFormat{title}{#1\addot\addthinspace}
\DeclareFieldFormat{journalttitle}{#1\isdot}%添加\isdot用于缩写名带点情况，将其转换为缩写点，便于标点的追踪
\DeclareFieldFormat{issuetitle}{#1}
\DeclareFieldFormat{maintitle}{#1}
\DeclareFieldFormat{booktitle}{#1}
\DeclareFieldFormat%将期刊等文献的标题中原来带的引号去掉
[article,patent,thesis,unpublished]
{title}{#1\addot\addthinspace}%\mkbibquote{#1\isdot}
\DeclareFieldFormat%将期刊等文献的标题中原来带的引号去掉
[inbook,incollection,inproceedings]
{title}{#1}%\nopunct\unspace

%
% 标题的字母大小写格式修改
%
% 注意:修改标题的字母大小写，不能用前面的title的格式而要用titlecase
% 因为titlecase is applied to the contents of the field directly, title is not
%\DeclareFieldFormat{titlecase}{\MakeCapital#1}%重设标题格式，将其修改为首字母大写
\DeclareFieldFormat{titlecase}{\iftoggle{bbx:titlelink}{%
\iffieldundef{url}{\MakeCapital#1}{%
\href{\thefield{url}}{#1}}{\MakeCapital#1}}}%重设标题格式，将其修改为首字母大写

%
% 重设title的输出
%
% 20180425, v1.0k, 为标题增加字体控制命令, Hu Zhenzhen
% 原理方法:将文献类型标识符输出出去，原输出来自biblatex.def文件
% 利用toggle做标识符是否输出的判断
\renewbibmacro*{title}{%
\ifboolexpr{%
test{\iffieldundef{title}}%
and
test{\iffieldundef{subtitle}}%
}%
{}%
{\printtext[ttitle]{\bibtitlefont%增加字体控制命令
\printfield[ttitlecase]{title}%
\ifboolexpr{test {\iffieldundef{subtitle}}}%这里增加了对子标题的判断，解决不判断多一个点的问题
}{\setunit{\subtitlepunct}%
\printfield[ttitlecase]{subtitle}}%
\iffieldundef{titleaddon}{}%判断一下titleaddon，否则直接加可能多一个空格
{\setunit{\subtitlepunct}\printfield{titleaddon}}%
\iftoggle{bbx:gotype}{%
\iffieldundef{note}{\printfield[gotypeflag]{usera}}%在标题后直接给出文献标识字母，判断一下，是否是报纸
和标准
{\iffieldequalstr{note}{standard}{\printfield[gotypeflags]{usera}}}%判断是否为标准
{\iffieldequalstr{note}{news}{\printfield[gotypeflagn]{usera}}}%判断是否
为报纸
{\printfield[gotypeflag]{usera}}}%其它
}}}%
%\iffieldundef{booktitle}{\newunit}{}%当title是析出时，不要标点
%\newunit
}%
}}

```

### 3.1.5.9 域输出的限制 利用选项 gbstrict 控制。

```
%
% 增加一个严格按GB/T 7714-2015给出著录格式控制域的输出的选项
% v1.0k, 2018.05.08, by hzz
%
% 原理方法:
% 默认设置该选项为真, 进而不输出GB/T 7714-2015中没有的域的信息
% 这样可以避免因为bib文件带有多余的域的信息的输出, 比如因为bibtex样式
% 可能需要用于中英文判断的language域的信息。
\newtoggle{bbx:gbstrict}
\DeclareBibliographyOption{gbstrict}[true]{%
  \settoggle{bbx:gbstrict}{#1}}
\ExecuteBibliographyOptions{gbstrict}
```

这个选项的机制是默认执行选项为 true, 进而设置 bbx:gbstrict 这一 toggle 为 true, bbx 中处理各条目类型的驱动根据 toggle 限制一些域的输出。当设置为 false 后, 则取消相应的限制。

### 3.1.6 著录格式控制

著录格式不仅包括整个文献表的格式, 还包括表中每一个条目的具体格式。下面分别介绍:

**3.1.6.1 文献表环境和打印输出** biblatex 宏包的参考文献表的打印与文献表环境相关, 或者说文献表环境直接决定了整个文献表的格式, 下面来分析 biblatex 中关于文献表环境的代码:

- (1) 根据 biblatex.sty 文件中的 `\blx@bibliography` 命令可以知道, 定义的参考文献表的环境的开始代码在命令中 `\blx@bibliography` 中, 结束代码在 `\blx@endbibliography` 中, 循环项代码在 `\blx@bibitem` 中的 `\blx@item@{\blx@theenv}` 命令中。参考文献循环打印依靠 `\blx@listloop` 命令实现。
- (2) 从 `\blx@listloop` 可以知道, 各条参考文献表的指引信息是混合在一起的并且以 | 字符为分隔。将混合的参考文献指引信息解析成单独的指引信息, 并交给 `\blx@bibitem` 命令处理。
- (3) `\blx@bibitem` 首先开始一个编组, 然后根据单独一条参考文献的指引信息, 获取数据, 执行判断, 利用参考文献表环境定义的循环项代码对参考文献内容设置段落格式, 最后结束编组。在定义 `gbalgn=gb7714-2015` 样式的段落格式时, 采用了 `parshape` 或者 `hangindent` 命令来实现, 所以其中增加了一个 `par` 命令以使其生效。

**3.1.6.2 作者格式控制** 【注意】: `texlive2015` 中的 `biblatex` 版本是 3.0, `texlive2016` 中 `biblatex` 的版本是 3.4, 新版本对于名字域打印有了较大变化 (即不同版本的 `biblatex` 对于 `DeclareNameFormat` 的输入参数处理有所不同, 如例 87 所示), 所以需做相应的修改, 为此在 `biblatex` 中首先进行版本判断, 然后根据版本不同分别进行设置。

示例 87 `texlive2016` 中 `biblatex3.4` 版 Name 域格式输入参数的修改

代码

```
1 for biblatex version 3.0
2 #1 The last names.
3 #2 The last names, given as initials.
4 #3 The first names.
```

```

5 #4 The first names, given as initials.
6 #5 The name prefixes,
7 #6 The name prefixes, given as initials.
8 #7 The name affixes,
9 #8 The name affixes, given as initials.
10 for biblatex version 3.4
11 \namepartfamily
12 \namepartfamilyi
13 \namepartgiven
14 \namepartgiveni
15 \namepartprefix
16 \namepartprefixi
17 \namepartsuffix
18 \namepartsuffixi

```

处理处理见样式文件。

作者信息的输出最高一层是在驱动中:

```
\usebibmacro{author/editor+others/translator+others}
```

其中当存在作者时, 由如下宏输出:

```
\renewbibmacro*{author}
```

其中作者列表又由如下命令输出:

```
\printnames{author}
```

printnames 这种命令的输出格式通常由相应输出域的域格式控制, 这里则是 author 域控制:

```
\DeclareNameAlias{author}{sortname}%for author year style
```

```
\DeclareNameAlias{author}{default}%for numeric style
```

其中 sortname, default 域格式又是:

```
\DeclareNameAlias{sortname}{family-given/given-family}
```

```
\DeclareNameAlias{author}{default}
```

```
\DeclareNameAlias{default}{given-family}
```

其中的关键是 given-family 和 family-given 两个域格式使用 name:family-given 和 name:family-given 宏进行输出。真实的作者姓名格式在这两个宏中, 因此做一定的处理。注意其中的输入参数为: namepartfamily, namepartgiveni, namepartprefix, namepartsuffix (使用名的首字母缩写) 或 namepartfamily, namepartgiven, namepartprefix, namepartsuffix (不使用缩写)。

%biblatex3.3版后(比如texlive2016中的3.4版)的使用方式

```

\renewbibmacro*{name:given-family}[4]{%利用family-given定义given-family
\ifuseprefix
{
\usebibmacro{name:delim}{##3##1}%
\usebibmacro{name:hook}{##3##1}%
\ifdefvoid{##3}{%
\ifcapital
{\mkbibnameprefix{\MakeCapital{##3}}\isdot}
{\mkbibnameprefix{##3}\isdot}%
\ifprefchar{}{\bibnamedelimc}}%
\ifdefvoid{##2}{\mkbibnamefamily{\MakeCapital{##1}}{\mkbibnamefamily{\MakeUppercase{##1}}\isdot}
\ifdefvoid{##2}{\revsdsnamepunct\bibnamedelimd\mkbibnamegiven{\MakeUppercase{##2}}\isdot}%\
\MakeCapital

```

```

\ifdefvoid{##4}{\bibnamedelimd\mkbibnamesuffix{##4}\isdot}}
{\usebibmacro{name:delim}{##1}%
\usebibmacro{name:hook}{##1}%
\ifdefvoid{##2}{\mkbibnamefamily{\MakeCapital{##1}}}{\mkbibnamefamily{\MakeUppercase{##1}}}\isdot
}%
\ifboolexpe{%
test {\ifdefvoid{##2}}
and
test {\ifdefvoid{##3}}
}{
{\revsdnamepunct}%
\ifdefvoid{##2}{\bibnamedelimd\mkbibnamegiven{\MakeUppercase{##2}}\isdot}%\MakeCapital
\ifdefvoid{##3}{\bibnamedelimd\mkbibnameprefix{##3}\isdot}
\ifdefvoid{##4}{\bibnamedelimd\mkbibnamesuffix{##4}\isdot}}
}
\renewbibmacro*{name:family-given}[4]{%
\usebibmacro{name:given-family}{##1}{##2}{##3}{##4}}

```

%biblatex3.3版前(比如texlive2015中的3.0版)的使用方式

```

\renewbibmacro*{name:last-first}[4]{%
\ifuseprefix
{\usebibmacro{name:delim}{##3##1}%
\usebibmacro{name:hook}{##3##1}%
\ifblank{##3}{\}%
\ifcapital
{\mkbibnameprefix{\MakeCapital{##3}}\isdot}
{\mkbibnameprefix{##3}\isdot}%
\ifpunctmark{'}{\bibnamedelimc}%
%\mkbibnamelast{#1}\isdot
\ifblank{##2}{\MakeCapital##1}{\mkbibnamelast{\MakeUppercase{##1}}\isdot}%\MakeUppercase %\
mkbibnamelast{\MakeUppercase{##1}} %\MakeSentenceCase
%注意上一句\MakeCapital后面如果再跟一个{}包含#1, 则没有效果, 可能是包在里面少了一层展开
%因为机构名通常包括在{}内, 所以要多展开一次才行, 所以这里去掉#1 外面的{}
%\mkbibnamelast{\MakeUppercase{##1}}\isdot
%\ifblank{##4}{\bibnamedelimd\mkbibnameaffix{##4}\isdot}%这句放到后面
\ifblank{##2}{\revsdnamepunct\bibnamedelimd\mkbibnamefirst{\MakeUppercase{##2}}\isdot}%
\ifblank{##4}{\addcomma\addspace\bibnamedelimd\mkbibnameaffix{##4}\isdot}}
{\usebibmacro{name:delim}{##1}%
\usebibmacro{name:hook}{##1}%
%\mkbibnamelast{##1}\isdot %3.9.1 Generic Commands and Hooks, 对姓重新处理, 如下句: %\mkbibnamelast{\
MakeUppercase{##1}}
\ifblank{##2}{\MakeCapital##1}{\mkbibnamelast{\MakeUppercase{##1}}\isdot} %大写, 参考4.6.4
Miscellaneous Commands, \MakeUppercase %\MakeSentenceCase
%\mkbibnamelast{\MakeUppercase{##1}}\isdot
%\ifblank{##4}{\bibnamedelimd\mkbibnameaffix{##4}\isdot}%这句放到后面
\ifblank{##2##3}{\revsdnamepunct}%
\ifblank{##2}{\bibnamedelimd\mkbibnamefirst{\MakeUppercase{##2}}\isdot}%
\ifblank{##3}{\bibnamedelimd\mkbibnameprefix{##3}\isdot}%
\addcomma\addspace%
\ifblank{##4}{\addcomma\addspace\bibnamedelimd\mkbibnameaffix{##4}\isdot}%
}
}

\renewbibmacro*{name:first-last}[4]{%
\usebibmacro{name:last-first}{##1}{##2}{##3}{##4}}

```

**3.1.6.3 超出数量的作者列表处理** 作者数量超出制定数量时, 需要用等或 et al. 来标识。处理宏如下:

```
%
% 修改作者数量超过限定值, 做省略时的处理格式
%
% 原理方法: 判断作者或译者是否中文, 若中文用字符等, 否则用etc1.
\renewbibmacro*{name:andothers}{%
  \ifboolexpr{%
    test {\ifnumequal{\value{listcount}}{\value{liststop}}}
    and
    test \ifmorenames
  }%
  {%这里做一个判断是在处理author还是translator用于两者是不同语言的情况
  \ifcurrentname{translator}{\testCJKfirst{usere}}{\testCJKfirst{userf}}}%
  %这句判断如果放到\andothersdelim后面会在等或etc.前增加一个空格, 所以放前面
  \ifnumgreater{\value{liststop}}{1}%
    {\finalandcomma}%
    {%biblatex作者要区别单作者加等的情况, 这里为符合gbt7714-2015第7.2节的要求加上了逗号。
  \printdelim{andothersdelim}\iftoggle{ifCJKforgbt}{\bibstring{andotherscn}}{\bibstring{andothers}}}%
  %\andothersdelim\bibstring{andotherscn}
}%
{}}
```

**3.1.6.4 日期格式控制** 随着 biblatex 的更新, 日期格式控制更趋完善。加上 biblatex3.8a 中 biber 解析出日期中月份和天数都不带引导的 0, 比如 6 月直接解析为 6, 而不像以前那样解析为 06, 所以需要特别的注意。因此对于新版本, 有些日期比如 online 的发布日期, 访问日期等, 采用专门的格式选项如 edtf (v3.10 后弃用)、iso 等进行控制, 而老版本则仍然使用原来定义的输出宏。比如:

```
\defversion{3.7}{date}{
  \DeclareFieldFormat{urldate}{##1}
  \renewbibmacro*{urldate}{%
    \addspace\printtext{[]}\printurldate\printtext{[]}%能用高层命令+选项尽量用命令(比如这里的\printurldate),
    而不用\blx@edtfdate这种更底层的命令

    \newbibmacro*{newsdate}{%%新增一个新闻日期
    \blx@edtfdate{}{}}%
  }

  \newbibmacro*{modifydate}{%新增一个带括号的日期, 用于表示电子资源的更新和修改日期, 而公告日期则按日期格式
    \iffielddundef{year}{%
      \iffielddundef{endyear}{\iffielddundef{eventyear}}{\printtext{()}\printeventdate\printtext{}}}%
      %
      {\printtext{()}\printenddate\printtext{}}}%
    }{\iffieldequalstr{year}{}}{%因为year存在, 但为空
      {\printtext{()}\blx@edtfdate{}{}}\printtext{()}}%
    }%
  }%
}

\defversion{3.4}{date}{
  \renewbibmacro*{urldate}{%
    \addspace\printtext{[]}\printfield{urlyear}%
    \iffielddundef{urlmonth}{\bibbrangedash\printfield{urlmonth}}}%
    \iffielddundef{urlday}{\bibbrangedash\printfield{urlday}}\printtext{[]}}
```



```
\iftoggle{iftlfive}%%根据texlive/biblatex版本选择
{\switchversion{3.4}{date}}%
{\iftoggle{iftlsix}%
{\switchversion{3.4}{date}}%
{\switchversion{3.7}{date}}%
}
```

### 3.1.6.5 条目标签中的日期格式

主要用于 `authoryear` 样式，在 `authoryear` 样式中有一个 `mergedate` 选项，使用该选项可以进行更多的标签日期格式控制。其中主要是重定义了 `date+extradate` (3.8 版，以前的版本叫 `date+extrayear`)。

其中有两点需要注意:

1. `printdateextra`, `printlabeldateextra`, 这样的命令在 `biblatex.sty` 中是找不到的定义, 因为 `biblatex` 实际有定义的命令是这样的:

{\protected\def\blx@imc@printlabeldate\csuse{mkdaterange#1}{label}}  
\protected\def\blx@imc@printlabeldateextra\csuse{mkdaterange#1extra}{label}}}

printdateextra, printlabeldateextra 等命令是由如下命令解析为上述命令:

```
\bblx@regimcs{\printlabeldate \printlabeltime \printlabeldateextra \stripzeros \forcezerosy \forcezerosmdt
\mkyearzeros \mkmonthzeros \mkdayzeros \mktimezeros}
```

`biblatex` 很多命令都有类似的方式。

2. 日期的数据来源由 `DeclareLabeldate` 定义选择。日期的具体格式，则由宏包选项控制。比如 `labeldate=ymd` 来控制。(3.5 版以上，以前的是 `datelabel` 选项)

**3.1.6.6 版本项格式控制** version/edition 域的做法格式处理，增加中文判断函数，并增加了注释符以避免带入空格。

```
%对version的版本信息做出修改，来自biblatex.DEF
\DeclareFieldFormat{version}{%
\testCJKfirst{userd}%userd保存有标题的信息，用于中文判断
\ifinteger{#1}%
{\iftoggle{ifCJKforgbt}{\printtext{#1版}}%
{\mkbibordedition{#1}~\bibstring{version}}}%
{#1\isdot}}
```

**3.1.6.7 卷期域格式控制** 卷期范围解析函数用于处理连续出版物，2 个卷期的情况，函数如下：

```
%用于解析连续出版物，2个卷期的情况
%\def\multivolumeparsetoparts#1/#2{\def\multivolumefirst{#1}\def\multivolumesecond{#2}}
%\def\multinumberparsetoparts#1/#2{\def\multinumberfirst{#1}\def\multinumbersecond{#2}}
%范围起止间隔符号还是用-，而不是与date相同的/，因为有合期期刊的问题，需要用到/符号
\newcommand{\multivolparser}[1]{%
\IfSubStr{#1}{-}%
{\StrBefore{#1}{-}[\multivolfirst]\StrBehind{#1}{-}[\multivolsecond]}%
{\def\multivolfirst{#1}\def\multivolsecond{}}}%
}

\newcommand{\multinumberparser}[1]{%
\IfSubStr{#1}{-}%
{\StrBefore{#1}{-}[\multinumberfirst]\StrBehind{#1}{-}[\multinumbersecond]}%
{\def\multinumberfirst{#1}\def\multinumbersecond{}}}%
}
```

%这里利用李志奇的范围数据解析函数，修改后用来处理卷volume的范围。

%他的这个函数是通用的，只能在使用数据前使用，如果多次使用前一次得到的数据就会被冲掉

%这里的范围起止判断符号还是用-，而不是与date相同的/，因为有合期期刊的问题，需要用到/符号

```
%\def\gbt@parse@range#1{%
% \edef\gbt@tmpa{#1}%
% \expandafter\gbt@parse@rangei\gbt@tmpa-\@empty}
%\def\gbt@parse@rangei#1-#2\@empty{%
% \global\def\gbt@range@first{#1}%
% \def\gbt@tmpa{#2}%
% \ifx\gbt@tmpa\@empty%
% \global\def\gbt@range@second{}%
% \else%
% \gbt@parse@rangeii#2\@empty%
% \fi}
%\def\gbt@parse@rangeii#1-#2\@empty{%
% \def\gbt@tmpa{#2}%
% \ifx\gbt@tmpa\@empty%
% \global\def\gbt@range@second{#1}%
% \else%
% \gbt@parse@rangeii#2\@empty%
% \fi}
```

%这里写的number的解析函数，与上面的卷解析函数是类似的。

```
%\def\multinumberparser#1{%
% \edef\gbnumbertmp{#1}%
% \expandafter\multinumberparsetoparts\gbnumbertmp-\@empty}
%\def\multinumberparsetoparts#1-#2\@empty{%
% \def\multinumberfirst{#1}%
```

```
% \def\gnumberbtmp{#2}%
% \ifx\gnumberbtmp\@empty%
% \def\multinumbersecond{}}%
% \else%
% \multinumberparsetopartsb#2\@empty%
% \fi}
%\def\multinumberparsetopartsb#1-#2\@empty{%
% \def\gnumberbtmp{#2}%
% \ifx\gnumberbtmp\@empty%
% \def\multinumbersecond{#1}%
% \else%
% \multinumberparsetopartsb#2\@empty%
% \fi}
```

### 3.1.6.8 标点和空格的特点和机制 用于实现满足标准要求的标点和间距。

- (1) 标点符号设置比如: `renewrobustcmd*{\bibinitperiod}{}{} 和 renewcommand*{\revsnamepunct}{}{} 等用于全局修改标点。`
- (2) 如 `isdot/adddot`, `adddot` 将原样 (逐字, 如实) 句点转换为缩写点在有利用 `printtext` 输入原样字符的时候很重要比如 `[s.n.]` 中, 比如 `title` 末尾。
- (3) 利用 `newunit` 输出的标点, 需要后面遇到 `printfield` 等命令有内容才输出标点, 这是标点的异步处理机制。
- (4) `setunit*` 的作用是前面 `printtext` 等没有输出时候就不输出, 见 4.11.7.3 节的应用以及出版项缺省时的代码处理。
- (5) 标点和空格的调试方法: 有时引入多余的空格和标点后很难快速处理, 那么需要调试, 调试的第一步是注释掉多余的代码, 实现无多余空格或标点情况 (一种快速方法是直接在一个 `printtext` 输出各相关域), 然后一步步恢复原来的代码, 直到找到多余空格或标点产生的原因, 最后根据原因作出修改。
- (6) 多余空格的引入, 可能来自于如下方面 (下面是已经遇到的, 可能还有更多)
  - 行末空格引入, 解决方法: 把相关的代码行结尾用 `%` 符号注释。
  - `mkbibparens` 等命令引入, 解决方法: 利用 `printtext{({}, printtext{)}} 来代替`
  - `nopunct` 等命令引入, 这种引入的空格使用 `unspace` 命令还消除不了, 解决方法: 避免使用 `nopunct` 命令, 而在标点设置时多做判断, 只有有内容时才设置标点, 而不是过分依赖 `biblatex` 提供的标点异步处理机制。
- (7) 短横线相关的字符, 参考: `unicode` 表
  - `0x2d:hyphen :-`, 直接输入键盘短横线得到 `-`, 即为连字符
  - `0x2010:hyphen :-`
  - `0x2011:non breaking hyphen :-`
  - `0x2012:figure dash :`
  - `0x2013:en dash :-`, 直接输入两个键盘短横线得到 `--`
  - `0x2014:em dash :一`, 中文全角的一段破折号线, 可以用三个短横线得到 `---`
  - `0x2212:mathematical minus :-`

`biblatex` 中页码范围中的间隔符是 `\bibbrangedash`, 默认则是 `en dash`。可以通过设置如下命令做修改:

```
\DefineBibliographyExtras{english}{\renewcommand*{\bibbrangedash}{-}}% 将页码间隔符替换为
hyphen连字符
```

而日期成分之间的间隔符则是`\bibdatesep`，默认是 `hyphen`，即连字符。

- (8) 使用 `printtext` 等命令确保异步处理机制不被破坏，典型的例子为，专利的公告日期 (对于 3.7 以上版本) 使用了 `biblatex` 的内容宏，但还需要将其用 `printtext` 包围起来，比如：

```
%
% 专利的公告日期、或报纸的日期的输出宏
% 20160701, v1.0, 新增加
% 20180405, 为biblatexv3.10版本, 出现多出点bug做处理, Hu Zhenzhen
%
% 原理方法:加上printtext避免破坏异步标点机制
%
\newbibmacro*{newsdate}{%%
\printtext{\blx@isodate{}{}}%%
}
```

- (9) `beamer` 中标题后面的标点情况，`beamer` 对 `bibmacro{title}` 做了 `patch` 以实现特殊的效果。比如：

```
\pretocmd{\abx@macro@title}
{
\ifcsundef{abx@name@labelname}{\let\bbx@tempa\labelnamepunct}%
\bbx@tempa\newblock\unspace\usebeamercolor[fg]{bibliography entry title}}{}{}
\apptocmd{\abx@macro@title}
{
\ifcsundef{abx@field@title}{\ifpunct{}{\midsentence\newunitpunct}}
\newblock\unspace\usebeamercolor[fg]{bibliography entry note}}{}{}
}
```

该 `patch` 详见 `beamerbaselocalstructure.sty` 文件。需要注意 `abx@field@title`，在 `biblatex` 中是有 `newbibmacro{title}` 产生的。注意到在 `apptocmd` 做的 `patch` 中，`midsentence` 命令隐藏了其前面的标点并重新做标点最终，因此其后的 `newunitpunct` 命令必将输出标点，因此无论如何标题后面必然会出现一个 `newunitpunct` 控制的标点，这个标点一般情况下是可以用的，但是对于有些文献类型比如 `inbook`，析出文献标题后面没有标点只有 `//`，因此需要将该标点去掉，所以需要反处理该 `patch`。通过测试表明，无论是在 `title` 前再做 `pretocmd` 或者在 `title` 后再做 `apptocmd` 都会产生问题，多个的 `apptocmd` 的机制大体是前一个做完后，后一个在其基础上处理，而 `apptocmd` 主要做的是封装，因此前一个产生的效果无法消除。但通过思考发现，使用 `patchcmd` 可以对前面的效果产生影响，它不是封装而是一种替换，所以可以将前面封装的效果去掉。比如：

```
%
% 文献标题后的标点问题
% 20180405, v1.0k, 为texlive2017以上版本中的beamer兼容性做的处理, Hu Zhenzhen
%
% 原理方法:如下代码处理texlive2017以上版本中, beamer中文献的标题后出现两个点的情况:
% texlive2017以上的beamer中对macro{title}做了patch, 正常情况下不会出现两个点的情况, 但由于
% 之前为了处理texlive2015, 2016下的title格式添加了adddot, 导致出现两个点的情况, 而且也影响
% 析出文献的//符号的输出, 因此再次对macro{title}做patch消除beamer中做apptocmd时添加的\
newunitpunct
\ifbool{prf}{
test{\iftoggle{iftlfive}}
or
test{\iftoggle{iftlsix}}
}{}%
{%texlive 2017对应iftlseven以上版本
\DeclareFieldFormat{title}{#1}%
\DeclareFieldFormat{article,patent,thesis,unpublished}{title}{#1}}
```

```

\AtBeginDocument{%
\ifclassloaded{beamer}
{\patchcmd{\abx@macro@title}{\newunitpunct}{}{}{}}{}
}

```

### 3.1.7 动态数据修改

利用 biber 在处理数据源时的动态处理, 可以处理一些数据, 比如设置一些域的值用于进一步的判断和应用; 也可以用来进行中英文判断; 也可以用来处理关联条目等等。

**3.1.7.1 基本原理和方法** biber 动态数据处理, 详见 biblatex 说明文档 Dynamic Modification of Data 一节和 bbx 文件中 DeclareSourceMap 的内容。

- (1) map 的作用, 对条目逐条进行处理
- (2) final 作用, 当不成立 map 终止。
- (3) append 的作用, 给域添加信息
- (4) origfieldvalue 来源
- (5) overwrite 选项有无的作用, 无 overwrite 时, 只要域原有信息, 那么就不再 map, 有 overwrite 时, 则进行覆盖。**注意: 使用 append 的时候也需要 overwrite 选项。**
- (6) biblatex 3.4 以后的版本可以使用 foreach 选项, 而 3.0 版只能一个域一个域的处理, 意味着每一个域处理都要写一个 map 步。
- (7) date 域仅作为解析用, 不出现在 bbl 文件中, 所以在 bbx 内部进行日期判断的时候要注意, 不使用 date 域而要用由其解析出来的 year 等域进行判断, 而且当 date 域的内容不符合解析格式要求时, 自动忽略掉, 那么信息有可能丢失, 因此当有不符合解析格式的日期信息时应放到 year 域中。
- (8) 动态数据处理中利用正则表达式可以用来处理特殊字符, 对于参考文献信息中一些特殊字符比如 & 等, 除了利用 jabref 软件的 biblatex 可以自动转换外, 利用动态数据修改也可以做一定的处理, 比如对一些容易出现这种字符的域进行处理, 使其内容符合 tex 源文件书写规则, 即在特殊字符前加上斜杠。注意利用正则表达式处理 %, # 时直接在 regexp 中写字符没有问题, 但是对于 & 字符就存在问题, 所以对于该字符使用十六进制表示方法 \x26。比如:

```

%for texlive >2016
\DeclareStyleSourceMap{
\maps[datatype=bibtex]{
\map[overwrite, foreach={title,booktitle,journaltitle,journal,publisher,address,
location,institution,organization}]{
\step[fieldsource=\regexp{$MAPLOOP}, match=\regexp{([^\])\#}, replace=\regexp{
$1\\#}]
}
\map[overwrite, foreach={title,booktitle,journaltitle,journal,publisher,address,
location,institution,organization}]{
\step[fieldsource=\regexp{$MAPLOOP}, match=\regexp{([^\])\%}, replace=\regexp{
$1\\%}]
}
\map[overwrite, foreach={title,booktitle,journaltitle,journal,publisher,%
address,location,institution,organization}]{
\step[fieldsource=\regexp{$MAPLOOP}, match=\regexp{([^\])\x26}, replace=\regexp{
$1\\x26}]
}
}
}

```

```

%for texlive 2015
\DeclareStyleSourceMap{
  \maps[datatype=bibtex]{
    \map[overwrite]{%title,booktitle,journaltitle,journal,publisher,address,location,
      institution,organization
    \step[fieldsource={title}, match=\regex{([^\])\x26}, replace=\regex{$1\\\x26}]
    }
    \map[overwrite]{
    \step[fieldsource={booktitle}, match=\regex{([^\])\x26}, replace=\regex{$1\\\x26}]
    }
    \map[overwrite]{
    \step[fieldsource={journaltitle}, match=\regex{([^\])\x26}, replace=\regex{$1\\\x
      26}]
    }
    \map[overwrite]{
    \step[fieldsource={journal}, match=\regex{([^\])\x26}, replace=\regex{$1\\\x26}]
    }
    \map[overwrite]{
    \step[fieldsource={publisher}, match=\regex{([^\])\x26}, replace=\regex{$1\\\x26}]
    }
    \map[overwrite]{
    \step[fieldsource={address}, match=\regex{([^\])\x26}, replace=\regex{$1\\\x26}]
    }
    \map[overwrite]{
    \step[fieldsource={location}, match=\regex{([^\])\x26}, replace=\regex{$1\\\x26}]
    }
    \map[overwrite]{
    \step[fieldsource={institution}, match=\regex{([^\])\x26}, replace=\regex{$1\\\x
      26}]
    }
    \map[overwrite]{
    \step[fieldsource={booktitle}, match=\regex{([^\])\#}, replace=\regex{$1\\\#}]
    }
    \map[overwrite]{
    \step[fieldsource={booktitle}, match=\regex{([^\])\%}, replace=\regex{$1\\\%}]
    }
  }
}

```

### 3.1.7.2 数据模型 下面介绍 biblatex 数据模型与动态数据处理的关系。

biblatex 提供了数据模型定义功能，这在真正需要定义一些新的条目类型和域时是用得到的。但实际上，从本样式实践的经验看，为了 biblatex 的兼容性考虑，没有必要使用数据模型定义功能去费力地定义新的条目类型，反而直接利用动态数据处理来将 bib 中的一些类型转换成 biblatex 支持的类型会更方便，其中要做一定的区分可以利用一些域的信息进行判断，比如 note 域等于 news 或 standard。

- (1) biblatex 默认做的别名处理其实是用 DeclareDriverSourceMap 做的，这是驱动层的映射。在本样式中，因为 usera 域是在用户层的映射中做的，所以 biblatex 做的默认别名处理还不够，还需要在用户层首先处理 usera 域。而因为 DeclareDriverSourceMap 中会增加一些不需要的信息，比如 thesis 类型中的 typer 域在 GB/T7714 标准中是并不需要的，这可以利用 DeclareSourceMap, DeclarestyleSourceMap(注意 texlive2016 的 biblatex3.4 仅支持 1 个 styleSourceMap,而 biblatex3.7 则已经



支持多个 `styleSource`map) 先进行映射, 这样就避开了 `DeclareDriverSource`map。或者在增加 `DeclareDriverSource`map, 将标准样式中的默认 `DeclareDriverSource`map 增加的信息去掉 (详见 `gb7714-2015ay.bbx` 中的处理和说明)。

```
\DeclareStyleSourceMap{
  \maps[datatype=bibtex]{
    % \map{%尝试未定义数据模型的newspaper类型映射为article, newspaper完全是针对gb7714的新类型, 在
    % biblatex中完全没有定义
    % %但从实践看, 并没有什么影响, 映射过来就可以了。这一段可以用下面的驱动层别名映射替代, 所以这里注
    % 释掉用下面的\DeclareBibliographyAlias命令
    % \step[typesource=newspaper, typetarget=article, final]
    % }

    \map{%尝试未定义数据模型的standard类型映射为book, standard类型在blx-dm中有出现, 但仅定义
    % 了类型, 域和约束等都没有定义
    \step[fieldsource=booktitle,final]%当存在booktitle域是映射为inbook
    \step[typesource=standard, typetarget=inbook, final]
    }
    \map{%尝试未定义数据模型的standard类型映射为book, standard类型在blx-dm中有出现, 但仅定义
    % 了类型, 域和约束等都没有定义
    \step[typesource=standard, typetarget=book, final]%当不存在booktitle 域是映射为book
    }
    \map{%先于标准样式的driver层映射, 将其先映射过来, 并取消type设置
    \step[typesource=mastersthesis, typetarget=thesis, final]
    %\step[fieldset=type, fieldvalue=mathesis]
    }
    \map{%先于标准样式的driver层映射, 将其先映射过来, 并取消type设置
    \step[typesource=phdthesis, typetarget=thesis, final]
    %\step[fieldset=type, fieldvalue=phdthesis]
    }
    \map{%先于标准样式的driver层映射, 将其先映射过来, 并取消type设置
    \step[typesource=techreport, typetarget=report, final]
    %\step[fieldset=type, fieldvalue=techreport]
    }
  }
}

\DeclareBibliographyAlias{newspaper}{article}%定义驱动别名, 尝试以替代驱动层映射, 实践表明是可
行的。
```

- (2) 数据模型在 `blx-dm.DEF` 中, 可以看到 `standard` 有条目类型定义, 但域和约束没有定义, 而 `biblatex` 标准样式默认做的别名处理中的其它一些条目比如 `electronic` 等是有域和约束定义的。
- (3) 标准样式 `standard.BBX` 中的 `DeclareBibliographyAlias{*}{misc}` 将一些数据模型中没有定义的条目直接用 `misc` 条目的驱动处理, 这不是驱动层的数据源映射, 只是利用其驱动进行输出。有一个问题是 `DeclareBibliographyAlias{newspaper}{article}` 是否可以实现? 还是要定义数据模型? 从实践结果看是不需要定义数据模型的, 并且驱动的别名处理等价于驱动层映射。比如: 做了用户层映射后:

```
\DeclareSourceMap{
  \maps[datatype=bibtex]{%增加文献标识符如[M],[J]等, 下面首先生成这一信息
    \map{
      \pertype{newspaper}%增加一个新闻报纸的类型newspaper
      \step[fieldset=usera, fieldvalue={N}]%因为没有专门的驱动, 这句的目的是定义一个
      % usera域, 方便映射为article 后判断
      \step[fieldset=note, fieldvalue=news]
    }
  }
}
```

```
}
```

做别名处理:

```
\DeclareBibliographyAlias{newspaper}{article}%定义驱动别名，以替代驱动层映射
```

能达到驱动层或样式层中条目类型转换所实现的效果。

```
\DeclareStyleSourceMap{
  \maps[datatype=bibtex]{
    \map{%尝试未定义数据模型的newspaper类型映射为article，newspaper完全是针对gb7714的
      新类型，在biblatex 中完全没有定义
      %但从实践看，并没有什么影响，映射过来就可以了。
    \step[tources=newspaper, typetarget=article, final]
  }
}
```

(4) 从实践看，数据模型中完全没有定义的条目类型 newspaper 也可以进行数据源映射。

**3.1.7.3 用于关联条目的动态修改的命令封装及关联条目格式** 实现动态的条目关联，可以利用动态数据修改的方法。要对多个条目实现关联，那么需要多次数据映射，biblatex3.7 及以上版本可以存在多个 `DeclareStyleSourceMap`，因此做关联的封装命令就是基于此：

```
\newcommand{\defdoublelangentry}[2]{%
\edef\entrykeya{#1}
\edef\entrykeyb{#2}
\DeclareStyleSourceMap{
  \maps[datatype=bibtex]{
    \map{
      \step[fieldsource=entrykey, match=\entrykeya, final]
      \step[fieldset=related, fieldvalue=\entrykeyb]
    }
  }
}
```

命令 `defdoublelangentry` 有两个输入参数，第一个是主条目，第二个是与主条目关联的条目的列表。多次使用该命令，即可实现多次数据映射，当然因为 `DeclareStyleSourceMap` 只能存在于导言区，因此 `defdoublelangentry` 也只能存在于导言区。

关联条目之间用命令 `relateddelim` 分隔，但实际上因为 `related` 宏输出时，不知道是何原因考虑的是主条目的关联条目之间的分隔，这在双语文献中是有问题的，需要在主条目后也进行分隔，因此对该宏修改如下：

```
\renewbibmacro*{related}{%
\ifboolexpr{ test {\iffieldundef{related}} or test {\ifrelatedloop} }
{}
{\usebibmacro{begrelated}%
\def\bbx@tempa{}%
\setcounter{bbx:relatedtotal}{0}%
\def\do##1{%
\entrydata{##1}{%
\ifrelatedloop
{}
{\stepcounter{bbx:relatedtotal}%
\gappto{\bbx@tempa}{##1,}}}%
\docsvfield{related}%
\restorefield{related}{\bbx@tempa}%
}
```

```

\ifnumgreater{\value{bbx:relatedtotal}}{0}
{
  \listcsxadd{bbx:relatedloop}{\strfield{entrykey}}%
  \iffieldundef{clonesourcekey}
  {}
  {\listcsxadd{bbx:relatedloop}{\strfield{clonesourcekey}}}%
  \setcounter{bbx:relatedcount}{0}%
  \def\do{%
    \stepcounter{bbx:relatedcount}%
    \ifnumgreater{\value{bbx:relatedcount}}{0}%为使主条目与关联条目做分隔, 这里做修改, 从1改为0
    {\ifcsundef{relateddelim}\strfield{relatedtype}}
    {\printtext{\relateddelim}}
    {\printtext{\csuse{relateddelim}\strfield{relatedtype}}}}
    {}}%
  \ifbibmacroundef{related:\strfield{relatedtype}}
  {\appto{\do}{\usebibmacro{related:default}}}
  {\appto{\do}{\usebibmacro*{related:\strfield{relatedtype}}}}%
  \iffieldformatundef{related:\strfield{relatedtype}}
  {\def\bbx@tempa{related}}
  {\def\bbx@tempa{related:\strfield{relatedtype}}}%
  \iffieldformatundef{relatedstring:\strfield{relatedtype}}
  {\def\bbx@tempb{relatedstring:default}}
  {\def\bbx@tempb{relatedstring:\strfield{relatedtype}}}%
  \printtext[\bbx@tempa]{%
    \usebibmacro{begrelatedloop}%
    \iffieldundef{relatedstring}
    {\ifboolexpr{
      test {\ifnumgreater{\value{bbx:relatedtotal}}{1}}
      and
      test {\ifbibxstring{\thefield{relatedtype}s}}
    }
    {\printtext[\bbx@tempb]{%
      \bibstring[\mkrelatedstring]{\thefield{relatedtype}s}}
    {\iffieldbibstring{relatedtype}
      {\printtext[\bbx@tempb]{%
        \bibstring[\mkrelatedstring]{\thefield{relatedtype}}}}
      {}}}
    {\iffieldbibstring{relatedstring}
      {\printtext[\bbx@tempb]{%
        \bibstring[\mkrelatedstring]{\thefield{relatedstring}}}}
      {\printfield[\bbx@tempb]{relatedstring}}}%
    \docsvfield{related}%
    \usebibmacro{endrelatedloop}}}%
    {}%
  \usebibmacro{endrelated}}}

```

### 3.1.8 标注样式的设计途径

针对 GB/T 7714-2015 标注, 设定相应的选项, 其中考虑不同 biblatex 版本带来的变化。同时也修改或增加用于实现标准要求的标注命令。

标注命令定义使用 `DeclareCiteCommand` 命令, 设置上标, 包围符号, 标注前后内容等。针对 GB/T 7714-2015 中关于引用标注的特殊要求, 增加了一些方便实现要求的命令比如 `pagescite`、`yearpagescite` 等。具体的实现见 `gb*.cbx` 文件。

**3.1.8.1 标签中的标点和间隔符号** 文献表中的标签和正文中的标注标签都有标点和间隔符号问题, 但两者可能存在一定的差异, 不同版本的 biblatex 中也可能不同, 但新的版本中通常是可以用一个相同的钩子来实现的, 比如作者年制中作者和年份之间的标点:

```
\renewcommand*{\nameyeardelim}{\addcomma\space}
```

**3.1.8.2 标注 (引用) 标签中的作者数量** 引用标签的生成机制, 即责任者截短时的歧义消除问题, 见说明文档 4.11.4 Name Disambiguation 节。

- (1) uniquename 的作用, 用于姓名间的歧义消除。
- (2) uniquelist 的作用, 用于姓名列表间的歧义消除。

关于这两个选项的参数设置, 可以在样式中设置, 也可以在宏包加载时设置, **注意宏包加载时的设置会覆盖样式中的设置。**

还要注意, 说明文档中 uniquelist 的默认选项是 false, 这是错误的, 从实践看是 true, 从 biblatex.sty 文件的代码看也是如此, 比如:

```
\DeclareBibliographyOption[boolean]{uniquelist}[true]{%
  \ifcsdef{blx@opt@uniquelist@#1}
    {\letcs\blx@uniquelist{blx@opt@uniquelist@#1}}
    {\blx@err@invopt{uniquelist=#1}{}}}
\DeclareTypeOption[boolean]{uniquelist}[true]{%
  \ifcsdef{blx@opt@uniquelist@#1}
    {\letcs\blx@uniquelist{blx@opt@uniquelist@#1}}
    {\blx@err@invopt{uniquelist=#1}{}}}
\DeclareEntryOption[boolean]{uniquelist}[true]{%
  \ifcsdef{blx@opt@uniquelist@#1}
    {\letcs\blx@uniquelist{blx@opt@uniquelist@#1}}
    {\blx@err@invopt{uniquelist=#1}{}}}
\def\blx@opt@uniquelist@false{0}
\def\blx@opt@uniquelist@true{1}
\def\blx@opt@uniquelist@myear{2}
```

无论选项是否给出, 上面定义选项中的代码都会自动执行, 当选项不给出, 则自动调用默认的参数 (因为代码中直接使用了输入参数, 当没有输入参数, 自然就会使用默认参数) 来进行处理, 显然这里是 true。

## 3.2 国标文档的完整复现

参考: tgbcitation, tgbcitationay 等。

一个简单示例: 对某个 cite 命令的输出做修改。这里采用直接修改该命令的输出宏的方式进行。另一种思路是对当前的 cite 命令做判断, 然后重定义输出宏即可。

```
\documentclass{article}
\usepackage{xCJK}
\usepackage[style=gb7714-2015]{biblatex}
\addbibresource{ref.bib}
\begin{filecontents}[force]{ref.bib}
@book{xxx,
  author = {蒋有绪 and 郭泉水 and 马娟 and others},
  title = {中国森林群落分类及其群落特征},
  address = {北京},
  publisher = {科学出版社},
  year = {1998},
}

@book{yyy, author = {Ammar Sohail and Arif Hidayat and Muhammad Aamir Cheema and David Taniar},}

\end{filecontents}
```

```

\DeclareDelimFormat[cite,parencite,textcite,citeauthor*]{andothersdelim}{}

\DeclareNameFormat{namefmtwithoutother}{%
  \ifgiveninits
    {\usebibmacro{name:family}
      {\namepartfamily}
      {\space}
      {\namepartprefix}
      {\namepartsuffix}}
    {\usebibmacro{name:family}
      {\namepartfamily}
      {\space}
      {\namepartprefix}
      {\namepartsuffix}}}%
}

\DeclareCiteCommand*{\citeauthor}
  {\boolfalse{citetracker}%
   \boolfalse{pagetracker}%
   \usebibmacro{prenote}}
  {\ifciteindex
    {\indexnames{labelname}}
    {}%
   \printnames[namefmtwithoutother][1-1]{labelname}}
  {\multicitedelim}
  {\usebibmacro{postnote}}

\begin{document}
\citeauthor*{xxx}

\citeauthor*{yyy}

\cite{xxx}

\cite{yyy}

\parencite{xxx}

\textcite{xxx}

\citep{xxx}

\citet{xxx}

\printbibliography
\end{document}

```

### 3.3 一个项目示例

#### 3.3.1 介绍

这是一个国内某期刊英文版的参考文献定制样式，项目不是很复杂，给出了两个示例，一是文献表示例 (图43)，二是引用标签示例 (图44)。我们需要根据这两个示例来开发相应的样式，即 `bbx` 和 `cbx` 文件。

相比之下，这种只有有限示例，而没有像 GB/T7714-2015 标准那样详细规定的样式，开发起来会自由一些，因为只要实现示例中的格式即可，未定义的作者可以自己把握，而不需

- least 25. The references mentioned in the text should accord with the reference list. For a reference published other than in English, the language used should be noted at the end of the reference list, e.g., (in Chinese). The publisher and place of publication should be given for a book or proceedings.)<sup>43</sup>
- Miller TC, Tobin RL, Freisz TL, 1991. Stackelberg games on a network with Cournot-Nash oligopolistic competitors. *Journal of Regional Science*, 31(4), 435-454. (Journals)<sup>44</sup>  
DOI: xxx<sup>45</sup>
- Hino T, Martinelli L, Jameson A, 1993. A finite-volume method with unstructured grid for free surface flow simulations. *Proceedings of the 6th International Symposium on Numerical Ship Hydrodynamics*, Iowa City, USA, 173-193. (Proceedings)<sup>46</sup>
- Harker PT, 1987. *Predicting intercity freight flows*. VNU Science Press, Utrecht, the Netherlands, 20-25. (Whole books)<sup>47</sup>
- Prigogine I, 1976. Order through fluctuation: self-organization and social system. In: Jantsch E, Waddington C (Eds.). *Evolution and Consciousness: Human Systems in Transition*. Addison-Wesley, London, 93-134. (Monographs or chapters in edited books)<sup>48</sup>
- Cone CD, 1963. *The aerodynamic design of wings with cambered span having minimum induced drag*. Langley Research Center, Virginia, United States, NASA Technical Report No. TR R-152. (Reports)<sup>49</sup>
- Hsin C, 1990. *Development and analysis of panel methods for propellers in unsteady flow*. PhD thesis, Massachusetts Institutes of Technology, Cambridge, 15-20. (Thesis)<sup>50</sup>
- Carter RW, Eretkin RC, 2011. Induced surface flow wave energy converter. U.S. Patent 8084873 B2. (A patent)<sup>51</sup>
- ISO, 1982. ISO 4948-1:1982. Steels classification-Part 1: Classification of steels into unalloyed and alloy steels based on chemical composition. International Organization for Standardization, Geneva. (A standard)<sup>52</sup>
- University of Sheffield Library, 2001. Citing electronic sources of information. University of Sheffield. Available from <http://www.shef.ac.uk/library/libdocs/hsl-dvc1.pdf> [Accessed on Feb. 23, 2007]. (A website)<sup>53</sup>

图 43 定制项目文献表示例

Examples of text citations:<sup>54</sup>  
One author: (Vandermeer, 1990)<sup>55</sup>  
Two authors: (Sun and Wang, 2000; Cao and Xu, 2001)<sup>56</sup>  
Three or more authors: (Moons *et al.*, 1997; Schlag *et al.*, 2000a; 2000b)<sup>57</sup>

图 44 定制项目引用标签示例

要详细测试每一条规定。此外因为该格式只有英文文献，语言只有英文，也不存在多语言的问题。但该格式也有一些有挑战的东西，特别是引用标签与文献表的作者标签存在明显不同，对于使用相同钩子的两个标签，可能需要特别的处理，以实现不同的格式。

参考文献样式的定制前面已经有了初步介绍，详见2.9.2节，本项目开发也是类似的，关于定制的基本方法不再赘述，下面主要介绍定制需要的所有功能是如何实现的。

### 3.3.2 bibx 文件开发

如前所述，biblatex 样式文件开发方法，有一个非常便捷的方式是基于标准样式进行。这种方式不仅相对简单，而且对 biblatex 的兼容性也更好。当 biblatex 升级时，除非有 api 改动，绝大多数情况下可以不用修改即可复用。

在本项目开发中，可以看到参考文献表的样式与标准样式很像，因此完全可以在其基础上进行开发，同时还可以利用 gb7714-2015ay 样式，其中有不少符合中文规范的处理。本项目主要利用这两个样式的代码进行开发。

**3.3.2.1 文献表环境** 看文献表的第一眼可以观察到各条目的缩进格式，即第一行顶格，第二行开始缩进约 3 个字母宽度，这是由 biblatex 参考文献环境控制，通常是 list 环境构建，设置合适的参数就可以实现目标。

示例 88 重定义参考文献环境确定缩进

```
1 %=====
2 %参考文献列表环境
3 %=====
4 \defbibenvironment{bibliography}%修改参考文献列表环境，不要缩进
5 {\list
6 {}
```

代码



---

```

7      {%\setlength{\leftmargin}{\bibhang}%
8      %\setlength{\itemindent}{-\leftmargin}%
9      \setlength{\leftmargin}{1.5em}%
10     \setlength{\itemindent}{-1.5em}%
11     \setlength{\itemsep}{\bibitemsep}%
12     \setlength{\parsep}{\bibparsep}}
13 {\endlist}
14 {\item}

```

---

**3.3.2.2 作者域处理** 接着我们看示例中个条目类型相同的部分，主要是作者和页码。这里先处理作者域。显然，作者标签有点像中文风格，姓在前，名的缩写为大小首字母，且没有加点。firstname 和 givenname 之间没有空格，姓的后面也没有标点，姓的首字母大写，其它小写。这种格式与 gb7714-2015 的风格有点像，可以复用其代码，略作大小写修改，同时使用几个钩子将标点和空格处理完毕。

#### 示例 89 作者域处理

代码

---

```

1  \renewrobustcmd*{\bibinitperiod}{%
2  \renewcommand*{\revsnamepunct}{\space}
3  \renewcommand*{\bibnamedelimd}{%
4  \renewcommand*{\bibnamedelima}{%
5  \renewcommand*{\bibinitdelim}{%
6
7  \renewbibmacro*{author}{%
8    \ifboolexpr{
9      test \ifuseauthor
10     and
11     not test {\ifnameundef{author}}
12   }
13   {\usebibmacro{bbx:dashcheck}
14     {%\bibnamedash
15     \printnames{author}%
16     }%
17     {\usebibmacro{bbx:savehash}%
18     \printnames{author}%
19     \iffieldundef{authortype}
20     {\setunit{\addspace}}
21     {\setunit{\addcomma\space}}}%
22   \iffieldundef{authortype}
23   {}
24   {\usebibmacro{authorstrg}%
25   \setunit{\addspace}}}%
26   {\global\undef\bbx@lasthash
27   \usebibmacro{labeltitle}%
28   \setunit*{\addspace}}%
29   \setunit{\addcomma\space}%增加一个逗号
30   \usebibmacro{date+extrayear}
31   }
32
33 %biblatex3.3版后(比如texlive2016中的3.4版)的使用方式
34 \renewbibmacro*{name:given-family}[4]{%利用family-given定义given-family
35   \ifuseprefix
36     {\usebibmacro{name:delim}{#3#1}%
37     \usebibmacro{name:hook}{#3#1}%
38     \ifdefvoid{#3}{-}{%
39       \ifcapital

```

---

```

40     {\mkbibnameprefix{\MakeCapital{#3}}\isdot}
41     {\mkbibnameprefix{#3}\isdot}%
42     \ifprefchar{}\{\bibnamedelimc}\}%
43     \ifdefvoid{#2}\{\mkbibnamefamily{\MakeCapital{#1}}\}\{\mkbibnamefamily{\MakeSentenceCase{#1}}\}\isdot
44     \ifdefvoid{#2}\{\{\revsdnamepunct\bibnamedelimd\mkbibnamegiven{\MakeUppercase{#2}}\isdot}\%\MakeCapital
45         {#2}
46     \ifdefvoid{#4}\{\{\bibnamedelimd\mkbibnamesuffix{#4}\isdot}\}
47     {\usebibmacro{name:delim}{#1}%
48     \usebibmacro{name:hook}{#1}%
49     \ifdefvoid{#2}\{\mkbibnamefamily{\MakeCapital{#1}}\}\{\mkbibnamefamily{\MakeSentenceCase{#1}}\}\isdot
50     %
51     \ifboolexpe{%
52         test {\ifdefvoid{#2}}
53         and
54         test {\ifdefvoid{#3}}
55         {}
56         {\revsdnamepunct}%
57     \ifdefvoid{#2}\{\{\bibnamedelimd\mkbibnamegiven{\MakeUppercase{#2}}\isdot}\%\MakeCapital{#2}
58     \ifdefvoid{#3}\{\{\bibnamedelimd\mkbibnameprefix{#3}\isdot}
59     \ifdefvoid{#4}\{\{\bibnamedelimd\mkbibnamesuffix{#4}\isdot}\}
60 \renewbibmacro*{name:family-given}[4]{%利用family-given定义given-family
61 \ifuseprefix
62     {\usebibmacro{name:delim}{#3#1}%
63     \usebibmacro{name:hook}{#3#1}%
64     \ifdefvoid{#3}\{\}%
65     \ifcapital
66         {\mkbibnameprefix{\MakeCapital{#3}}\isdot}
67         {\mkbibnameprefix{#3}\isdot}%
68     \ifprefchar{}\{\bibnamedelimc}\}%
69     \ifdefvoid{#2}\{\mkbibnamefamily{\MakeCapital{#1}}\}\{\mkbibnamefamily{\MakeSentenceCase{#1}}\}\isdot
70     \ifdefvoid{#2}\{\{\revsdnamepunct\bibnamedelimd\mkbibnamegiven{\MakeUppercase{#2}}\isdot}\%\MakeCapital
71         {#2}
72     \ifdefvoid{#4}\{\{\bibnamedelimd\mkbibnamesuffix{#4}\isdot}\}
73     {\usebibmacro{name:delim}{#1}%
74     \usebibmacro{name:hook}{#1}%
75     \ifdefvoid{#2}\{\mkbibnamefamily{\MakeCapital{#1}}\}\{\mkbibnamefamily{\MakeSentenceCase{#1}}\}\isdot
76     %
77     \ifboolexpe{%
78         test {\ifdefvoid{#2}}
79         and
80         test {\ifdefvoid{#3}}
81         {}
82         {\revsdnamepunct}%
83     \ifdefvoid{#2}\{\{\bibnamedelimd\mkbibnamegiven{\MakeUppercase{#2}}\isdot}\%\MakeCapital{#2}
84     \ifdefvoid{#3}\{\{\bibnamedelimd\mkbibnameprefix{#3}\isdot}
85     \ifdefvoid{#4}\{\{\bibnamedelimd\mkbibnamesuffix{#4}\isdot}\}

```

---

此外，多个姓名之间用逗号分隔，最后一个姓名前不用 and，而使用逗号，可以通过使用钩子以及处理本地化字符串 and 来实现。

#### 示例 90 本地化字符串处理

---

```

1 \DefineBibliographyStrings{english}{
2 and = {\addcomma},%将第2和3人名见的and符号改成 逗号，用\finalnamedelim命令也可以定义，参见3.9.1节
3 byeditor = {\mkbibparens{Eds\addot\isdot}},
4 urlseen = {Accessed on},
5 january = {Jan\addot},
6 february = {Feb\addot},

```

代码

```

7  march = {Mar\adddot},
8  april = {Apr\adddot},
9  may = {May},
10 june = {June},
11 july = {July},
12 august = {Aug\adddot},
13 september = {Sept\adddot},
14 october = {Oct\adddot},
15 november = {Nov\adddot},
16 december = {Dec\adddot},
17 }

```

**3.3.2.3 条目类型及其使用的域** 接下来我们首先来看一下示例中所给出的条目类型 (表4) 及其所使用的域。

表 4 支持的条目类型

entry type		references
article	for	journals' paper
inproceedings	for	proceedings' paper
book	for	whole books
inbook	for	monographs or chapters in edited books
report	for	reports
thesis/phdthesis/masterthesis	for	thesis
patent	for	patents
standard	for	standards
online/www	for	websites

根据对各条目内容的分析可以得到各条目使用的域为:

示例 91 各条目使用的域

代码

```

1  article:
2  author,date or year.title.journaltitle,volume(number),pages.doi
3
4  inproceedings:
5  author,date or year.title.booktitle.publisher,address or location,pages
6
7  book:
8  author,date or year.title.edition.publisher,address or location,pages
9
10 inbook:
11 author,date or year.title.editor or bookauthor.booktitle.edition.publisher,address or location,pages
12
13 report:
14 author,date or year.title.publisher,address or location.type or number,pages
15
16 thesis/phdthesis/masterthesis:
17 author,date or year.title.type.institution,address or location,pages
18
19 patent:
20 author,date or year.title.number
21
22 standard:

```

```

23 author,date or year.title.translator.publisher,address or location,pages
24
25 online/www:
26 author,date or year.title.organization.url.urldate

```

**3.3.2.4 article 类型处理** article 条目中, 标题为正常字体, 期刊名为斜体, 卷为粗体, 期用圆括号包围, 页码前面的符号为逗号, 无引导词如 pp., 而且 doi 需要换行。

修改期刊名钩子, 卷期定义一个新的域格式, 调整页码的引导符, 修改 doi 格式, 可以处理完成。

#### 示例 92 article 条目处理

代码

```

1 \DeclareFieldFormat{journalttitle}{\mkbibitalic{#1}\isdot}%添加\isdot, 用于缩写名带点情况, 将其转换为缩写点便于标点追踪
2 \DeclareFieldFormat%将期刊等文献的标题中原来带的引号去掉
3 [article,patent,unpublished,incollection,inproceedings,inbook]
4 {title}{#1\addot\addthinspace}
5 \DeclareFieldFormat{titlecase}{\MakeCapital#1}%重设标题格式, 将其修改为首字母大写
6
7 \DeclareFieldFormat{pages}{#1}%去掉前面引导页码的pp.等字符
8
9 \DeclareFieldFormat{addnumflag}{%%增加一个number带括号的格式, 避免使用mkbibparens而引入一个不必要的空格
10 \nobreak\printtext{()}\nobreak #1\nobreak\printtext{}}
11 \DeclareFieldFormat{addvolflag}{\mkbibbold{#1}}
12
13 \renewbibmacro*{journal+issuetitle}{%
14 \usebibmacro{journal}%
15 \setunit*{\addspace}%
16 \iffieldundef{series}
17 {\setunit{\addcomma\addspace}}
18 {\newunit
19 \printfield{series}%
20 \setunit{\addcomma\addspace}}%
21 \usebibmacro{volume+number+eid}%
22 \setunit{\addcomma\addspace}%
23 \usebibmacro{issue+date}%
24 \setunit{\addcolon\space}%
25 \usebibmacro{issue}%
26 \newunit}
27
28 \newbibmacro*{journal}{%
29 \iffieldundef{journalttitle}
30 {}
31 {\printtext[journalttitle]{%
32 \printfield{journalttitle}%
33 \setunit{subtitlepunct}%
34 \printfield{journalsubtitle}}}}
35
36 %调整期刊卷和期的格式,
37 \renewbibmacro*{volume+number+eid}{%
38 \printfield[addvolflag]{volume}%
39 %\setunit*{\addot}%去掉点号
40 %\printfield{number}%
41 \iffieldundef{number}{}\{\printfield[addnumflag]{number}}%
42 %\iffieldundef{number}{}\{\printtext{\mkbibparens{\printfield{number}}}}% 增加一个圆括号
43 \iffieldundef{eid}{}\{

```

---

```

44 \setunit{\addcomma\space}%
45 \printfield{eid}}
46
47 \renewbibmacro*{doi+eprint+url}{%
48 \iftoggle{bbx:eprint}
49   {\usebibmacro{eprint}}
50   {}%
51 \newunit\newblock
52 \iftoggle{bbx:url}
53   {\usebibmacro{url+urldate}}
54   {}
55 \newunit\newblock
56 \iftoggle{bbx:doi}
57   {\iffielddundef{doi}{\printtext{\newline}}\printfield{doi}}%注意这里printtext的运用, 避免前面点的跳行
58   {}
59
60 %修改article的驱动, 期刊文章, 连续出版物中的析出文献的格式
61 \DeclareBibliographyDriver{article}{%
62 \usebibmacro{bibindex}%
63 \usebibmacro{begentry}%
64 \usebibmacro{author/translator+others}%
65 \setunit{\printdelim{nametitle}}\newblock
66 \usebibmacro{title}%
67 \newunit
68 \printlist{language}%
69 \newunit\newblock
70 \usebibmacro{byauthor}%
71 \newunit\newblock
72 \usebibmacro{bytranslator+others}%
73 \newunit\newblock
74 \printfield{version}%
75 \newunit\newblock
76 %\usebibmacro{in:}%
77 \usebibmacro{journal+issuetitle}%
78 \newunit
79 \usebibmacro{byeditor+others}%
80 \newunit
81 \usebibmacro{note+pages}%
82 \newunit\newblock
83 \iftoggle{bbx:isbn}
84   {\printfield{issn}}
85   {}%
86 \newunit\newblock
87 \usebibmacro{doi+eprint+url}%
88 \newunit\newblock
89 \usebibmacro{addendum+pubstate}%
90 \setunit{\bibpagerefpunct}\newblock
91 \usebibmacro{pageref}%
92 \newunit\newblock
93 \iftoggle{bbx:related}
94   {\usebibmacro{related:init}%
95   \usebibmacro{related}}
96   {}%
97 \usebibmacro{finentry}}

```

---

**3.3.2.5 inproceedings 类型处理** 标题为正常字体, 文集名为斜体, 需要出版项, 同时观察 book, report 类型可以发现, 出版项中, 出版者在前, 出版地在后, 中间逗号连接, 出版

者缺省时不输出。

修改文集名的钩子，修改出版项格式，页码由前面的 article 中做的全局处理实现。注意到文集名后出版项前的符号是逗号，但这里使用了点号，因为作者认为这样更合理，不同信息块之间通常是由点分隔，示例中绝大部分情况都是如此。当然用逗号分隔也没有任何难度，只需在驱动中出版项前加入一个逗号标点即可。

#### 示例 93 inproceedings 条目处理

代码

```

1 \DeclareFieldFormat{maintitle}{\mkbibemph{#1}}
2 \DeclareFieldFormat{booktitle}{\mkbibemph{#1}}
3 \DeclareFieldFormat%将期刊等文献的标题中原来带的引号去掉
4 [article,patent,unpublished,incollection,inproceedings,inbook]
5 {title}{#1\adddot\addthinspace}
6
7 \renewbibmacro*{publisher+location+date}{%
8 \printlist{publisher}
9 \iflistundef{location}{%\setunit*{\addcomma\space}}%
10 \printlist{location}%%
11 \setunit*{\addcomma\space}%
12 \usebibmacro{date}%
13 \newunit}
14
15 %会议论文格式修改
16 \DeclareBibliographyDriver{inproceedings}{%
17 \usebibmacro{bibindex}%
18 \usebibmacro{begentry}%
19 \usebibmacro{author/translator+others}%
20 \setunit{\printdelim{nametitleldelim}}\newblock
21 \usebibmacro{title}%
22 \newunit
23 \printlist{language}%
24 \newunit\newblock
25 \usebibmacro{byauthor}%
26 \newunit\newblock
27 %\usebibmacro{in:}%
28 \usebibmacro{maintitle+booktitle}%
29 \newunit\newblock
30 \usebibmacro{event+venue+date}%
31 \newunit\newblock
32 \usebibmacro{byeditor+others}%
33 \newunit\newblock
34 \iffieldundef{maintitle}
35 {\printfield{volume}%
36 \printfield{part}}
37 {}%
38 \newunit
39 \printfield{volumes}%
40 \newunit\newblock
41 \usebibmacro{series+number}%
42 \newunit\newblock
43 \printfield{note}%
44 \newunit\newblock
45 \printlist{organization}%
46 \newunit
47 \usebibmacro{publisher+location+date}%
48 \newunit\newblock
49 \usebibmacro{chapter+pages}%
50 \newunit\newblock

```

---

```

51 \iftoggle{bbx:isbn}
52   {\printfield{isbn}}
53   {}%
54 \newunit\newblock
55 \usebibmacro{doi+eprint+url}%
56 \newunit\newblock
57 \usebibmacro{addendum+pubstate}%
58 \setunit{\bibpagerefunct}\newblock
59 \usebibmacro{pageref}%
60 \newunit\newblock
61 \iftoggle{bbx:related}
62   {\usebibmacro{related:init}%
63    \usebibmacro{related}}
64   {}%
65 \usebibmacro{finentry}}

```

---

**3.3.2.6 book 类型处理** 书名为斜体, 出版项使用与 `inproceedings` 一致的全局处理。驱动无需修改使用标准样式提供的即可。

#### 示例 94 book 条目处理

代码

---

```

1 \DeclareFieldFormat%
2   [book,report,thesis]
3   {title}{\mkbibemph{#1}}

```

---

**3.3.2.7 inbook 类型处理** 章节名正常字体, 全书名为斜体, 同时全书还需要作者或编者, 章节和书名之间用 `in:` 表示, 出版项和页码使用前述的全局处理。设置书名钩子, 处理 `in` 的本地化字符串格式后, 最重要是是全书编者中跟着的提示信息 (Eds.)。这种提示信息常是由本地化字符串提供, 显然 `eds` 表示编者, 通常全书的作者还可以用 `bookauthor` 域表示, 但这里为了体现 `eds` 信息, 使用 `editor` 域, 于是我们需要对 `editor` 域输出格式做修改, 得到符合要求的结果。其中 `byeditor` 本地化字符串处理, 见例90。

#### 示例 95 inbook 条目处理

代码

---

```

1 \DeclareFieldFormat{maintitle}{\mkbibemph{#1}}
2 \DeclareFieldFormat{booktitle}{\mkbibemph{#1}}
3
4 \renewbibmacro*{in:}{%
5   \printtext{%
6     \mkbibitalic{\bibstring{in}}\intitlepunct}}
7
8 \renewbibmacro*{byeditor+others}{%
9   \ifnameundef{editor}
10    {}
11    {\printnames[byeditor]{editor}%
12     \setunit{\addspace}%
13     \usebibmacro{byeditor+othersstrg}%
14     \clearname{editor}%
15     \newunit}%
16 \usebibmacro{byeditorx}%
17 \usebibmacro{bytranslator+others}}
18
19 %修改inbook的驱动
20 \DeclareBibliographyDriver{inbook}{%

```

---



---

```

21 \usebibmacro{bibindex}%
22 \usebibmacro{begentry}%
23 \usebibmacro{author/translator+others}%
24 \setunit{\printdelim{nametitleledelim}}\newblock
25 \usebibmacro{title}%
26 \newunit
27 \printlist{language}%
28 \newunit\newblock
29 \usebibmacro{byauthor}%
30 \newunit\newblock
31 \usebibmacro{in:}%
32 \usebibmacro{bybookauthor}%
33 \newunit\newblock
34 \usebibmacro{byeditor+others}%
35 \newunit\newblock
36 \usebibmacro{maintitle+booktitle}%
37 \newunit\newblock
38 \printfield{edition}%
39 \newunit
40 \iffieldundef{maintitle}
41   {\printfield{volume}%
42     \printfield{part}}
43   {}%
44 \newunit
45 \printfield{volumes}%
46 \newunit\newblock
47 \usebibmacro{series+number}%
48 \newunit\newblock
49 \printfield{note}%
50 \newunit\newblock
51 \usebibmacro{publisher+location+date}%
52 \newunit\newblock
53 \usebibmacro{chapter+pages}%
54 \newunit\newblock
55 \iftoggle{bbx:isbn}
56   {\printfield{isbn}}
57   {}%
58 \newunit\newblock
59 \usebibmacro{doi+eprint+url}%
60 \newunit\newblock
61 \usebibmacro{addendum+pubstate}%
62 \setunit{\bibpagerefpunct}\newblock
63 \usebibmacro{pageref}%
64 \newunit\newblock
65 \iftoggle{bbx:related}
66   {\usebibmacro{related:init}%
67     \usebibmacro{related}}
68   {}%
69 \usebibmacro{finentry}}

```

---

**3.3.2.8 report 类型处理** 在 report 类中，标题为斜体，后面是出版项，和报告类型及编号。修改驱动调整出版项和报告类型的顺序。也要注意两者之间示例给出的逗号分隔符，这里作者仍然使用了点号，原因如前所述，不同信息块的分隔用点号，当然如果要改成逗号也非常简单。唯一需要注意的是，通常报告的出版者用 institution 域表示，所以对其做域 publisher 类似的处理。

## 示例 96 report 条目处理

代码

---

```

1 \DeclareFieldFormat%
2   [book,report,thesis]
3   {title}{\mkbibemph{#1}}
4
5 \renewbibmacro*{institution+location+date}{%
6   \printlist{institution}%
7   \iflistundef{location}{\setunit*{\addcomma\space}}%
8   \printlist{location}%
9   \setunit*{\addcomma\space}%
10  \usebibmacro{date}%
11  \newunit}
12
13 %report
14 \DeclareBibliographyDriver{report}{%
15   \usebibmacro{bibindex}%
16   \usebibmacro{begentry}%
17   \usebibmacro{author}%
18   \setunit{\printdelim{nametitle}}\newblock
19   \usebibmacro{title}%
20   \newunit
21   \printlist{language}%
22   \newunit\newblock
23   \usebibmacro{byauthor}%
24   \newunit\newblock
25   \printfield{version}%
26   \newunit
27   \printfield{note}%
28   \newunit\newblock
29   \usebibmacro{institution+location+date}%
30   \newunit\newblock
31   \printfield{type}%
32   \setunit*{\addspace}%
33   \printfield{number}%
34   \newunit\newblock
35   \usebibmacro{chapter+pages}%
36   \newunit
37   \printfield{pagetotal}%
38   \newunit\newblock
39   \iftoggle{bbx:isbn}
40     {\printfield{isrn}}
41     {}%
42   \newunit\newblock
43   \usebibmacro{doi+eprint+url}%
44   \newunit\newblock
45   \usebibmacro{addendum+pubstate}%
46   \setunit{\bibpagerefpunct}\newblock
47   \usebibmacro{pageref}%
48   \newunit\newblock
49   \iftoggle{bbx:related}
50     {\usebibmacro{related:init}%
51      \usebibmacro{related}}
52     {}%
53   \usebibmacro{finentry}}

```

---

**3.3.2.9 thesis 类型处理** 标题是斜体，带论文类型，后面是出版项和页码。标题修改使用钩子，出版项和页码使用前述的全局设置。主要是论文类型。目前没有做任何额外处理，因

为 biblatex 自动会将 phdthesis, masterthesis 转换为 thesis, 并设置 type。如果用户直接使用 thesis, 并且需要 type 信息, 那么需要在 type 域中输入类型是 phdthesis 或 masterthesis。

**3.3.2.10 patent 类型处理** 标题为正常字体, 后面跟专利号。无需任何处理。

**3.3.2.11 standard 类型处理** 因为 standard 类型在 biblatex 中没有定义, 因此需要增加一个 standard 类型的驱动, 类似于 book 的驱动, 只是标题改为正常字体。

示例 97 standard 条目处理

代码

```

1 %增加标准的驱动
2 \DeclareBibliographyDriver{standard}{%
3   \usebibmacro{bibindex}%
4   \usebibmacro{begentry}%
5   \usebibmacro{author/editor+others/translator+others}%
6   \setunit{\printdelim{nametitle}{delim}}\newblock
7   \usebibmacro{maintitle+title}%
8   \newunit
9   \printlist{language}%
10  \newunit\newblock
11  \usebibmacro{byauthor}%
12  \newunit\newblock
13  \usebibmacro{byeditor+others}%
14  \newunit\newblock
15  \printfield{edition}%
16  \newunit
17  \iffieldundef{maintitle}
18    {\printfield{volume}%
19     \printfield{part}}
20    {}%
21  \newunit
22  \printfield{volumes}%
23  \newunit\newblock
24  \usebibmacro{series+number}%
25  \newunit\newblock
26  \printfield{note}%
27  \newunit\newblock
28  \usebibmacro{publisher+location+date}%
29  \newunit\newblock
30  \usebibmacro{chapter+pages}%
31  \newunit
32  \printfield{pagetotal}%
33  \newunit\newblock
34  \iftoggle{bbx:isbn}
35    {\printfield{isbn}}
36    {}%
37  \newunit\newblock
38  \usebibmacro{doi+eprint+url}%
39  \newunit\newblock
40  \usebibmacro{addendum+pubstate}%
41  \setunit{\bibpagerefpunct}\newblock
42  \usebibmacro{pageref}%
43  \newunit\newblock
44  \iftoggle{bbx:related}
45    {\usebibmacro{related:init}%
46     \usebibmacro{related}}
47    {}%
```

---

```
48 \usebibmacro{finentry}}
```

---

**3.3.2.12 online 类型处理** 最后是 online 类型, 标题后跟的是组织, 后面跟的是网址及其访问日期。主要要修改的是 url 和 urldate 的域格式。因为 urldate 域需要提供所有的年月日信息, 所以设置 urldate 宏包选项, 注意区别于 date 域的选项。同时因为其特殊的形式, 修改月份的本地化字符串, 如例90所示。

示例 98 online 条目处理

代码

---

```
1 \ExecuteBibliographyOptions{
2   useprefix = true, %名字的信息包括前缀
3   %firstinits = true, %名字有缩写, 参考3.1.2.3 Internal
4   giveninits = true,
5   date = year, %日期仅写到年
6   urldate= comp,
7   %maxnames = 3, %设置名字最大数量
8   %minnames = 3, %设置缩减后的名字最小数量
9   maxbibnames=3, %将文献列表和引用中最大名字数量区分开
10  minbibnames=3,
11 }
12
13 \DeclareFieldFormat{url}{Available from\space\url{#1}}
14 \DeclareFieldFormat{urldate}{\mkbibbrackets{\bibstring{urlseen}\space#1}}
```

---

**3.3.2.13 全局选项及 texlive 兼容性处理** 全局选项设置如例98所示, 使用类似于 gb7714-2015 的设置, 使用姓名前缀和缩写, 设置文献表中作者标签最大和最小姓名数为 3。

对于 biblatex 的兼容性, 因为升级可能带来的变化, 这里特别做了处理, 主要是 texlive2016 和 texlive2017, 其中的 biblatex 版本分别为 3.4 和 3.7。对于本项目, 其中有一个命令存在差异, 因此换个版本会报错, 因此特别做出处理, 处理方式类似于 gb7714-2015 样式。

示例 99 texlive 兼容性处理

代码

---

```
1 \providetoggle{iftexlivesix}
2 \providetoggle{iftexliveseven}%为处理texlive2016到2017的变化
3 %版本判断, 根据biblatex更新历史可知, 版本3.3开始使用新的姓名机制
4 %所以当版本大于3.2开始, 就用设置\toggletrue{iftexlivesix}
5 \def\numparserta#1.#2\relax{#1}%注意relax的重要性
6 \def\numparsertb#1.#2\relax{#2}
7 \def\numinteger{\expandafter\numparserta\abx@version\relax}
8 \def\numdigital{\expandafter\numparsertb\abx@version\relax}
9 \ifnumcomp{\numdigital}{>}{2}{\toggletrue{iftexlivesix}}{\togglefalse{iftexlivesix}}
10 \ifnumcomp{\numdigital}{>}{6}{\toggletrue{iftexliveseven}}{\togglefalse{iftexliveseven}}
11
12 \iftoggle{iftexliveseven}{%
13   \renewbibmacro*{date+extrayear}{%
14     \iffieldundef{labelyear}
15     {}
16     {\printtext{%[parens]}%这里把括号去掉
17       \iffieldsequal{year}{labelyear}
18       {\printlabeldateextra}%
19       {\printfield{labelyear}%
20         \printfield{extrayear}}}}}%
21 }
```

---

```

22 \renewbibmacro*{date+extrayear}{%
23   \iffieldundef{\thefield{datelabelsource}year}
24   {}
25   {\printtext{%[parens]%这里把括号去掉
26     \iffieldsequal{year}{\thefield{datelabelsource}year}
27     {\printdateextralabel}%
28     {\printfield{labelyear}%
29     \printfield{extrayear}}}}}%
30 }

```

如此完成了 bbx 文件的开发。

### 3.3.3 cbx 文件开发

下面进行 cbx 文件开发。cbx 引用标签样式示例给出了 3 个。分别是 1 位作者、2 位、3 位或以上作者的情况。可以看到 2 位作者时全部给出，超过 3 位则只给出一位，这可以通过设置宏包选项实现，例 100 中给出的其它选项，与 gb7714-2015 相同。单个姓名的标签采用默认的方式处理，作者和年份之间的分隔符，修改对应的钩子即可。

示例 100 cbx 设置宏包选项

```

1 \ExecuteBibliographyOptions{
2   %autocite = superscript ,
3   %autopunct = true ,
4   %sorting = none ,
5   maxcitenames=2,
6   mincitenames=1,
7   uniqueness=init,%因为使用了名字缩写选项，所以需要设置uniqueness=init而不是full避免冲突
8   labeldate=true,
9   uniquelist=minyear,
10 }
11
12 \DeclareDelimFormat{nameyear}{\addcomma\addspace}

```

代码

除此之外，还有几个格式需要实现，包括：

- et al. 的斜体
- 两位作者时，and 连接
- 同作者文献，连续出现，作者名只出现一次

这三个问题其实是本项目最复杂的地方。因为一般情况下，引用标签与文献表中的作者标签使用相同的钩子，因此格式相同。那么在文献表中作者格式确定的情况下，无法全局的重定义这些钩子，否则会影响文献表。

好在 tex 的作用域机制保证了重定义可以作用在局部范围内。于是重定义 name:andothers 宏来实现 et al 斜体，重定义 finalnamedelim 来实现 and 连接。但是作者省略的问题，还无法解决，思考解决方法，找到 authoryear 文献表中存在的省略，这里其实也可以复用其中的宏，于是新定义一个 cite:labelname 取代直接打印姓名的方式，重定义 cite 宏实现要求，最后重定义 cite 命令，在引用标签外加上圆括号。

示例 101 cbx 重定义实现要求

```

1 \newbibmacro*{cite:labelname}{%
2   \usebibmacro{bbx:dashcheck}%
3   {}%\bibnamedash%
4   {\usebibmacro{bbx:savehash}%

```

代码

---

```

5      \printnames{labelname}%
6      \setunit{\printdelim{nameyeardelim}}}}
7
8  \renewbibmacro*{cite}{%
9      \iffielddundef{shorthand}%
10     {\iffielddundef{labelname}\OR\iffielddundef{labelyear}}%
11     {\usebibmacro{cite:label}%
12     \setunit{\printdelim{nonameyeardelim}}}%
13     {\renewcommand*{\finalnamedelim}{%
14     \ifnumgreater{\value{liststop}}{2}{\finalandcomma}{}%
15     \addspace and\space}%
16     \renewbibmacro*{name:andothers}{%
17     \ifboolexpr{%
18     test {\ifnumequal{\value{listcount}}{\value{liststop}}}
19     and
20     test \ifmorenames
21     }%
22     {\ifnumgreater{\value{liststop}}{1}%
23     {\finalandcomma}%
24     {}}%
25     \andothersdelim\mkbibemph{\bibstring{andothers}}}%
26     {}}%
27     \usebibmacro{cite:labelname}%\printnames{labelname}%
28     }%
29     \usebibmacro{cite:labelyear+extrayear}}%
30     {\usebibmacro{cite:shorthand}}}}
31
32 %\DeclareCiteCommand{\cite}[\mkbibparens]{precode}{loopcode}{sepcode}{postcode}
33 %
34 \DeclareCiteCommand{\cite}[\mkbibparens]
35 {\usebibmacro{prenote}}
36 {\usebibmacro{citeindex}%
37 \usebibmacro{cite}}
38 {\multicitedelim}
39 {\usebibmacro{postnote}}

```

---

到此，我们完成了所有的开发。

项目所有文档见: [egzazhi.tex](#), [zazhi.bbx](#), [zazhi.cbx](#), [egbib.bib](#)。

结果如图45所示。

### 3.3.4 小结

通过上述实践，我们完成了一个定制项目，可以看到 biblatex 的强大，它提供的 api 相当全面和有效，利用它以及标准样式可以较为方便构建满足需要的定制格式。

## 3.4 样式定制中的一些技巧

### 3.4.1 选项的范围限制

选项有不同的作用范围，包括 Load-time、Global、Per-refcontext、Per-type、Per-entry、Per-namelist、Per-name 等，有时可以针对不同的需要来设置不同的选项作用范围，例就是一种针对条目类型的选项设置：

示例 102 针对条目类型的选项设置

代码

---

```

1 \ExecuteBibliographyOptions[%
2   book,%
3   bookinbook,%

```

## 1 bibliography

one author(Harker, 1987) (Hsin, 1990; Prigogine, 1976) (University of Sheffield Library, 2001; Cone, 1963; ISO, 1982)

two authors(Carter and Eretkin, 2011)

three or more authors(Hino *et al.*, 1993; Miller *et al.*, 1991a; 1991b)

## References

Carter RW, Eretkin RC, 2011. Induced surface flow wave energy converter. U.S. Patent 8084873 B2.

Cone CD, 1963. *The aerodynamic design of wings*. Langley research center, Virginia, USA. NASA technical reptot No. TR R-152.

Harker PT, 1987. *Predicting intercity freight flows*. VNU science Press, Utrecht, the Netherlands, 20–25.

Hino T, Martinelli L, Jameson A, 1993. A finite-vomlume method with unstructed grid for free surface flow simulations. *Proceedings of the 6th international symposium on numerical ship hydrodynamics*. Iowa City, USA, 173–193.

Hsin C, 1990. *Development and analysis of panel mmethod for propellers in unsteady flow*. PhD thesis. Massachusetts institutes of technology, Cambridge, 15–20.

ISO, 1982. ISO 4948-1:1982. Steels classification-Part 1:steels based on chemical composition. standard. International organization for standardization, Geneva.

Miller TC, Tobin RL, Freisz TL, 1991a. Stackelberg games on a network with C-N competitors. *Journal of regional science*, **31**(4), 435–454.  
DOI: [infoofdoi](#).

Miller TC, Tobin RL, Freisz TL, 1991b. Stackelberg games on a network with CN network competitors. *Journal of regional science*, **31**(4), 435–454.  
DOI: [infoofdoi](#).

Prigogine I, 1976. Order through fluctuation:self-organization and social system. In: Jantash E, Waddington C (Eds.). *Evolution and consciousness*. Addison-Wesley, London, 93–134.

University of Sheffield Library, 2001. Citing Electronic source of information. University of Sheffield. Available from <http://www.rfla.org/IV/rfla64/13801613.htm> [Accessed on Feb. 23, 2007].

1

图 45 样式定制结果

```

4 bookinarticle,%
5 bookincollection,%
6 bookinincollection,%
7 mvbook,%
8 inbook,%
9 bookininbook,%
10 bookininarticle,%
11 inreference,%
12 bookinreference,%
13 bookininreference%
14 ]{useeditor=false,usetranslator=false}

```

### 3.4.2 排序格式的宏重用

排序格式用 `DeclareSortingScheme`(老版本 `biblatex`)/`DeclareSortingTemplate` 定义, 很多时候可以利用宏重用来精简代码, 比如有多个不同的格式, 其中仅有部分的差别, 多数内容都是相同的, 此时可以将相同的内容定义为宏, 然后在不同的格式中重用, 比如:

示例 103 排序格式中的宏重用

```

1 \def\bbx@sorting@reused{
2   \sort{
3     \field{sortname}
4     \field{author}
5     \field{editor}
6     \field{translator}
7     \literal{0000}
8   }

```

代码



```

9   \sort{
10   \field{sorttitle}
11   \field{title}
12   }
13   \sort{
14   \field{sortyear}
15   \field{year}
16   }
17 }
18
19 \DeclareSortingScheme{anonymous}{
20   \sort{
21     \field{presort}
22   }
23   \sort[final]{
24     \field{sortkey}
25   }
26   \bbx@sorting@reused
27 }

```

### 3.4.3 不同位置/方式的代码处理

biblatex 提供了很多接口可以在不同的位置使用一些代码，比如在 `AtEndPreamble`、`AtEveryCitekey`、`AtDataInput` 等处，进行一些内容检测、筛选、替换等处理操作。比如，在 `AtEveryCitekey` 处检测数据来实现筛选替换。比如，在 `AtDataInput` 处对每个条目做检测以实现筛选。又比如，利用 `check`，利用 `map`，或者在宏中利用判断比如 `ifcitation` 等进行判断和后续操作等等。

具体代码，比如：

示例 104 不同位置的代码处理

代码

```

1  %导言区结束加代码：
2  \AtEndPreamble{
3    \ifundef{\bbx@manuscripts@sorting}{}%
4    {
5      \DeclareSortingScheme{anonymous+manuscripts}{
6        \sort{
7          \field{presort}
8        }
9        \sort[final]{
10         \field{sortkey}
11       }
12       \bbx@anonymous@sorting
13       \bbx@manuscripts@sorting
14     }
15   }
16 }
17
18 %条目引用开始处加代码：
19 \AtEveryCitekey{\iffieldundef{prenote}{\blx@initunit}{}}

```

示例 105 引用位置的本地化字符串/空格处理

代码

```

1  %本地化字符串：
2  %在正文标注中，无论中文作者还是英文作者都使用“等”作为作者缩略时的标记
3  %区别于文献表中，中文作者用等，而英文作者用 et al.

```

---

```

4 \AtEveryCite{%
5 \renewbibmacro*{name:andothers}{%
6 \ifboolexpr{
7 test {\ifnumequal{\value{listcount}}{\value{liststop}}}
8 and
9 test \ifmorenames
10 }{\ifnumgreater{\value{liststop}}{1}
11 {\finalandcomma}%
12 {}%\finalandcomma
13 \printdelim{andothersdelim}\bibstring{andotherscn}%
14 }{}}
15
16
17 %空格处理：
18 %引用每一篇文献时，局部定义andothersdelim分隔符
19 \AtEveryCitekey{%
20 \iffieldequalstr{userf}{cn}{\renewcommand*{\andothersdelim}{\addthinspace}}%\addthinspace
21 {\renewcommand*{\andothersdelim}{\addspace}}%
22 }
23 %而因为textcite命令中有专门的分隔符环境可以用，因此会覆盖andothersdelim的设置
24 \DeclareDelimFormat[textcite]{andothersdelim}{\iffieldequalstr{userf}{cn}{\addthinspace}{\addspace}}%

```

---

### 3.4.4 数据模型定义

biblatex 提供了数据模型的高级定义功能，可以自定义条目类型、域、域集合、条目类型支持的域等，比如下例定义了一种新的条目类型 newsarticle：

示例 106 数据模型定义

---

```

1 \DeclareDatamodelEntrytypes{
2 newsarticle}
3
4 \DeclareDatamodelFields[type=list, datatype=name]{
5 with,
6 writer,
7 producer,
8 execproducer,
9 director}
10
11 \DeclareDatamodelFields[type=field, datatype=literal]{
12 abstractloc,
13 titleisdescription,
14 newspaper,
15 urldescription}
16
17 \DeclareDatamodelFields[type=field, datatype=verbatim]{
18 abstracturl}
19
20 \DeclareDatafieldSet{author}{
21 \member[field=author]
22
23 \DeclareDatamodelEntryfields{
24 titleisdescription,
25 with,
26 abstractloc,
27 abstracturl,
28 urldescription}
29
30 \DeclareDatamodelEntryfields[newsarticle]{

```

代码

---

```

31  author,
32  newspaper,
33  title,
34  day,
35  endday,
36  endmonth,
37  endyear,
38  month,
39  year,
40  addendum,
41  doi,
42  editor,
43  editora,
44  editorb,
45  editorc,
46  editoratype,
47  editorbtype,
48  editorctype,
49  eid,
50  eprint,
51  eprintclass,
52  eprinttype,
53  issn,
54  issue,
55  issuetitle,
56  issuesubtitle,
57  language,
58  note,
59  number,
60  origlanguage,
61  pages,
62  series,
63  subtitle,
64  titleaddon,
65  translator,
66  url,
67  urlday,
68  urlendday,
69  urlendmonth,
70  urlendyear,
71  urlmonth,
72  urlyear,
73  volume}

```

---

当然, 数据模型定义很多时候并不是唯一的解决方案, 一些情况下, 利用条目类型别名设置已经能够满足需求, 不需要专门为某一类型条目设置数据模型。

### 3.4.5 日期格式控制选项定义

根据日期格式选择的定义可以知道, 样式作者可以为不同的日期增加新的选项, 先看一下 biblatex 中日期相关选项的定义:

示例 107 日期相关选项的定义

代码

---

```

1  \def\do#1{%
2  \DeclareBibliographyOption[string]{#1date}{%
3  \ifstrequal{##1}{iso8601}
4  {\blx@warning@noline{'iso8601' date format specifier is deprecated.\MessageBreak
5  Use 'iso' instead}%

```

```

6      \csdef{blx@dateformat@#1date}{iso}}
7      {\ifstrequal{##1}{edtf}
8        {\blx@warning@noline{'edtf' date format specifier is deprecated.\MessageBreak
9          Use 'iso' instead}%
10         \csdef{blx@dateformat@#1date}{iso}}
11        {\csdef{blx@dateformat@#1date}{##1}}}%
12  \ifstrequal{##1}{ymd}
13    {\renewrobustcmd*{\bibdaterangesep}{\slash}%
14     \renewrobustcmd*{\bibtimerangesep}{\slash}%
15     \renewrobustcmd*{\bibdateeraendprefix}{\bibdateeraprefix}}
16    {}%
17  \ifcsdef{mkdaterange##1}
18    {\protected\csdef{blx@imc@print#1date}{\noexpand\csuse{mkdaterange##1}{#1}}%
19     \protected\csdef{blx@imc@print#1dateextra}{\noexpand\csuse{mkdaterange##1extra}{#1}}}
20    {\ifcsdef{mkbibrange##1}%LEGACY(<3.5)
21     {\blx@warning@noline{%
22       '\string\mkbibrange##1(extra)' date range macro in style is deprecated,\MessageBreak
23       please define '\string\mkdaterange##1(extra)' instead}%
24      \protected\csdef{blx@imc@print#1date}{\noexpand\csuse{mkbibrange##1}{#1}}%
25      \protected\csdef{blx@imc@print#1dateextra}{\noexpand\csuse{mkbibrange##1extra}{#1}}}
26     {\blx@err@invopt{#1date=##1}{}}}%
27  % Default for new date fields is "comp" but this is overridden below by
28  % biblatex defaults
29  \setkeys{blx@opt@pre}{#1date=comp}%
30  % Time format options
31  \DeclareBibliographyOption[string]{#1time}{%
32    \csdef{blx@timeformat@#1time}{##1}%
33    \ifcsdef{mktimerange##1}
34      {\protected\csdef{blx@imc@print#1time}{\csuse{mktimerange##1}{#1}}}
35      {\blx@err@invopt{#1time=##1}{}}}%
36  % Default for new time fields is "24h"
37  \setkeys{blx@opt@pre}{#1time=24h}%
38  % Boolean to determine if times are printed with dates
39  \newtoggle{blx@#1dateusetime}%
40  \DeclareBibliographyOption[boolean]{#1dateusetime}{%
41    \ifstrequal{##1}{true}
42      {\toggletrue{blx@#1dateusetime}}
43      {\togglefalse{blx@#1dateusetime}}}%
44  \abx@dodatatypes

```

可以看到当存在`mkdaterange##1`命令时, `biblatex` 就会定义一个对应的`blx@imc@print#1date`用来输出日期。因此只要定义`mkdaterange##1`就可以为日期选项比如 `date`, `urldate` 等增加新的选项。

比如:

#### 示例 108 日期选项的增加

```

1  \newcommand*{\mkdaterangeapalong}[1]{%
2    \blx@metadateinfo{#1}%
3    \iffielddundef{#1year}{}
4    {\datecircaprint
5     \ifstrequal{#1}{url}% URL dates are unlikely to be BCE ...
6     {\printtext{%
7       \mkbibdateapalongmdy{#1year}{#1month}{#1day}%
8       \iffielddundef{#1endyear}%
9       {}%
10      {\iffieldequalstr{#1endyear}{}% open-ended range?

```

代码

```

11     {\mbox{\bibdatedash}}
12     {\bibdatedash%
13       \iffieldsequal{#1year}{#1endyear}%
14       {\iffieldsequal{#1month}{#1endmonth}%
15         {\iffieldsequal{#1day}{#1endday}%
16           {}%
17           {\mkbibdateapalongmdy{}{}{#1endday}}}%
18         {\mkbibdateapalongmdy{}{#1endmonth}{#1endday}}}%
19         {\mkbibdateapalongmdy{#1endyear}{#1endmonth}{#1endday}}}%
20       \dateuncertainprint}}
21   {\printtext{%
22     \mkbibdateapalong{#1year}{#1month}{#1day}%
23     \dateeraprint{#1year}%
24     \iffieldundef{#1endyear}%
25     {}%
26     {\iffieldequalstr{#1endyear}{}% open-ended range?
27       {\mbox{\bibdatedash}}
28       {\bibdatedash%
29         \iffieldsequal{#1year}{#1endyear}%
30         {\iffieldsequal{#1month}{#1endmonth}%
31           {\iffieldsequal{#1day}{#1endday}%
32             {}%
33             {\mkbibdateapalong{}{}{#1endday}}}%
34           {\mkbibdateapalong{}{#1endmonth}{#1endday}}}%
35           {\mkbibdateapalong{#1endyear}{#1endmonth}{#1endday}%
36             \dateeraprint{#1endyear}}}%
37         \enddateuncertainprint}}}}}}
38
39   \protected\def\mkbibdateapalongmdy#1#2#3{%
40     \iffieldundef{#2}%
41     {}%
42     {\mkbibmonth{\thefield{#2}}}%
43     \iffieldundef{#3}%
44     {}%
45     {\addspace}%
46     \stripzeros{\thefield{#3}}}%
47     \iffieldundef{#1}%
48     {}%
49     {\ifthenelse{\iffieldundef{#3}}%
50       {\ifthenelse{\iffieldundef{#2}}%
51         {}%
52         {\addspace}}%
53       {\addcomma\addspace}%
54     \iffieldbibstring{#1}{\biblcstring{\thefield{#1}}{\thefield{#1}}}}

```

### 3.4.6 美式标点

示例 109 美式标点

代码

```
1 \uspunctuation
```

### 3.4.7 不同条目类型的域格式

示例 110 不同条目类型的域格式

代码

```

1 \DeclareFieldFormat[article]{number}{\bibstring{number}\addnbspace#1}
2 \DeclareFieldFormat[patent]{number}{\mkonepagegrouped{#1}}
3 \DeclareFieldFormat{pages}{%
4   \mkpageprefix[bookpagination][\mkpagegrouped]{#1}%
5 }

```

---

```

6 \DeclareFieldFormat{sentencecase}{\bbx@colon@search{\MakeSentenceCase*}{#1}}
7 \DeclareFieldFormat[book,inbook,incollection,inproceedings]{series}
8   {\bibstring{jourser}\addnbspace#1}
9 \DeclareFieldFormat[online]{title}{#1\isdot}
10 \DeclareFieldFormat[report]{title}{\mkbibquote{#1\isdot}}
11 \DeclareFieldFormat[url]{\bibstring{url}\addcolon\space\url{#1}}
12 \DeclareFieldFormat*{volume}
13   {\bibstring{volume}\addnbspace#1}
14 \DeclareFieldFormat[article,periodical]{volume}
15   {\bibstring{jourvol}\addnbspace#1}

```

---

### 3.4.8 文献表中相同作者用 dash 代替

可以利用 `authoryear` 中的代码:

示例 111 相同作者用 dash 代替

代码

---

```

1 % An option that carries through from author-year styles
2 \DeclareBibliographyOption[boolean][dashed][true]{%
3   \ifstrequal{#1}{true}
4     {\ExecuteBibliographyOptions{pagetracker}%
5       \renewbibmacro*{bbx:savehash}{\savefield{fullhash}{\bbx@lasthash}}}
6     {\renewbibmacro*{bbx:savehash}{}}}
7
8 \InitializeBibliographyStyle{\global\undef\bbx@lasthash}
9
10 \newbibmacro*{bbx:savehash}{\savefield{fullhash}{\bbx@lasthash}}
11
12 % Bibliography macros
13 \renewbibmacro*{author}{%
14   \ifboolexpr{
15     test \ifuseauthor
16     and
17     not test {\ifnameundef{author}}
18   }
19   {%
20     \iffieldequals{fullhash}{\bbx@lasthash}
21       {\bibnamedash\addcomma\space}
22       {\printnames{author}}}%
23     \usebibmacro{bbx:savehash}%
24     \iffieldundef{authortype}
25       {}
26       {%
27         \setunit{\addcomma\space}%
28         \usebibmacro{authorstrg}%
29       }%
30   }
31   {\global\undef\bbx@lasthash}%
32 }

```

---

## 4 参考文献的传统解决方案

参考文献的传统解决方案, 在很多文档中其实有详细介绍, 比如 `lshort-zh-cn` 等, 用户能在其中获取足够的信息, 本节只做一些总结性的说明。

### 4.1 传统解决方案的基本原理

详见《1.1 latex 参考文献基本原理》一节的说明。

## 4.2 基于 thebibliography 环境的方法

最简单的插入参考文献的方式是用 thebibliography 环境, 只需要使用 thebibliography 环境手工输入。在文档末尾 (或在正文环境内任意位置) 插入该环境, 编译两遍即可。

thebibliography 环境显式的将其中的内容插入, 而不管其中的参考文献是否都有所引用。利用该环境可以生成多个参考文献, 并不会出现问题, 因此可以用于生成分章的参考文献。

顺序编码制示例见: [eg-thebibliography-num.tex](#), 著者年份制示例见: [eg-thebibliography-ay.tex](#), 注意到示例中利用了 natbib 宏包来生成相应样式的引用标注。

## 4.3 基于 bibtex 的方法

### 4.3.1 基本操作

基于 bibtex 的方法是一种将参考文献信息和源文档分开管理的方法。应用过程中涉及到三类要素: 源文档 (Tex 文件)、参考文献数据库 (bib 文件)、参考文献样式 (bst 文件)。参考文献数据库是参考文献的数据基础, 参考文献信息以条目形式录入其中, 由于参考文献数据库文件是纯文本的文件, 可以利用 wine<sub>dt</sub> 等文本编辑器直接写, 也可以利用各种工具获取。参考文献样式有 latex 及各类第三方宏包或提供, 用户只需要选择使用即可。在源文档中, 用户引用相关文献的关键字后, 可利用 bibliography<sub>style</sub> 和 bibliography 命令生成参考文献。

基于 bibtex 程序, 利用 bibliography<sub>style</sub> 和 bibliography 命令生成的参考文献可以与 thebibliography 环境的参考文献同时存在, 但是 bibliography<sub>style</sub> 和 bibliography 命令生成的参考文献无法利用一个 bib 文件生成多个文献 (可以插入多个但却是相同的), 而利用多个 bib 文件也不行, 因为编译文档时, 所有的信息都是写入一个 aux 文件中的。因此分章参考文献需使用 chapterbib 等宏包, 而且有一些预设条件比如要用 include 命令来引入不同的章节, 以产生多个 aux 文件, 到达生成多个文献表的目的。

### 4.3.2 文档编译

基于 bibtex 的方法编译时首先利用 latex 编译文档生成 aux 文件, 然后利用 bibtex 编译 aux 文件和 bib 文件, 最后再进行两次 latex 编译。通常是四次编译:

- (1) 调用 xelatex 对 tex 源文档做第一次编译, 扫描文中引用的参考文献, 将参考文献引用信息写入辅助 (aux) 文件。
- (2) 调用 bibtex, 以参考文献数据库文件、参考文献格式文件和辅助文件为输入, 生成格式化以后的参考文献著录和引用信息, 写入 bbl 文件。
- (3) 再次调用 xelatex 编译 tex 源文档, 将格式化以后的参考文献条目插入正文。
- (4) 再次调用 xelatex 编译 tex 源文档, 生成参考文献引用标签插入正文。

而 biblatex 完成相同任务一般仅需三次编译, xelatex、biber、xelatex。若还需要反向超链接, 则也需要进行四次编译。

一个简单示例如下:

示例 112 采用 bibtex 的文献生成示例

代码

```
1 \begin{filecontents*}{\jobname.bib}
2 @phdthesis{zhao2003,
3   author = {赵温波},
4   title = {径向基概率神经网络研究},
5   school = {中国科学技术大学},
6   year = {2003},
7   type = {[博士]},
```



---

```

8   address = {合肥},
9   pages={21-24}
10 }
11
12 @Inproceedings{Nemec1997-209-214,
13   Title = {Force control of redundant robots},
14   Author = {B Nemec},
15   Booktitle = {Processings of Symposium on Robot Control},
16   shortbooktitle = {(PSRC)},
17   Pages = {209-214},
18   Country = {Nantes France},
19   Year = {1997}
20 }
21
22 @Article{Chiani1998-2998-3008,
23   Title = {Error probability for block codes over channels with block interference},
24   Author = {Chiani, M.},
25   Journal = {IEEE Trans. Inf. Theory},
26   shortbooktitle = {(ITIT)},
27   Number = {7},
28   Pages = {2998-3008},
29   Volume = {44},
30   Year = {1998}
31 }
32 \end{filecontents*}
33
34 \documentclass{ctexart}
35 \usepackage{natbib}
36 \bibliographystyle{plainnat}
37
38 \begin{document}
39   \cite{zhao2003}、\citet{Nemec1997-209-214}、\citep{Chiani1998-2998-3008}
40
41   \bibliography{\jobname}
42 \end{document}

```

---

### 4.3.3 常用著录样式

BIBTEX 提供了几个预定义的样式，如 plain, unsrt, alpha 等，第三方宏包、论文模板等也会提供一些文献样式，比如 natbib 宏包提供的样式 plainnat, abbrvnat, unsrtnat，比如其它的一些样式 harvard, apalike, chicago, astron, authordate, ieetrans 等。国内最常用是实现国标 GB/T 7714 要求的样式，包括胡海星开发的 2005 样式<sup>[19]</sup>，以及 Lee Zeping 开发的 2015 样式<sup>[20]</sup>。

参考文献表的所有格式、逻辑细节都由著录样式实现，比如各个域（字段）的著录格式，文献条目见得排序逻辑等等。

### 4.3.4 常用标注（引用）样式

引用文献生成标注也可以有多种样式，例如直接使用 latex 的标注样式，也可以通过 natbib 等宏包进行自定义。目前选用 natbib 宏包的方法最为常用，可以利用其生成著者年份制的标注（使用 authoryear 选项），以及顺序编码标注（使用 numbers 或 super 选项）。

### 4.3.5 注意事项

总结一些注意事项：

- 参考文献只有在正文中被引用了，才会在最后的参考文献列表中出现，若需要生成未引用的文献，可使用 cite\* 命令。

- 使用参考文献数据库和样式时, 需要准确写明文件, 文件名不能出错, 且不需要扩展名, 比如 bib 文件是 ref.bib, 那么 bibliography 命令的参数是 ref, bib 文件名是 ref.tex.bib, 则参数应是 ref.tex。若著录样式为 plain.bst, 那么 bibliographystyle 命令的参数应为 plain。
- 只要 tex 源文档和 bib 数据源是以 utf-8 编码的, 那么传统参考文献方法中引用文献时都可以采用中文的 bibtex 键。例子见 eg-bibliography.tex。

#### 4.4 natbib 宏包用法简介

##### 4.4.1 基本概念

natbib 宏包主要用来生成标注标签, 在不同的章节可以使用不同的标注样式。

##### 4.4.2 用法和注意事项

1. 当与 chapterbib 联用时, sectionbib 选项需要在 natbib 宏包加载时给出。给 chapterbib 设置无效。

2. 文献表的样式仍然设置用 `\bibliographystyle{}` 命令。

3. natbib 提供了常用的 citet 和 citep 命令, 通常 citet 会提供作者信息, citep 会加上包围的符号。当然在 natbib 加载不同的选项时, 会有不同的表现。

4. natbib 提供了三个 plainnat.bst abbrvnat.bst unsrtnat.bst 样式文件。

5. thebibliography 环境的语法:

```
\bibitem[Jones et al.(1990)]{jon90}...
```

或

```
\bibitem[Jones et al.(1990)Jones, Baker,and Williams]{jon90}...
```

其中 (年份) 前的是缩略的姓名列。后面的则是详细的列表。该列表在首次引用需要详细姓名的样式中会用到。

6. 提供了大量的 cite 类命令, 可以输出作者, 年份, 数字等信息。可以大小写变化。这些按需使用, 最常用的还是前面提过的 citet 和 citep。

7. 还可以利用 `\defcitealias` 命令定义别名标签, 某些情况下要生成特定的标签会非常有用。

8. 可以利用 `\setcitestyle` 命令来局部设置标注的样式, 该命令的参数是 key=val 或 key 结构。也可以利用 `\bibpunct` 命令来设置标注的样式, 主要是标点。也可以利用 `\bibstyle@xxx` 命令来预设置一种标注样式, 然后用 `\citestyle` 命令来使用。

比如: `\newcommand{\bibstyle@agu}{\bibpunct{[]{}{};}{a}{,}{,}{~}}`

然后使用: `\citestyle{agu}`

natbib 已经预定义的样式包括: plain、plainnat、agu、egu、agms、cospar、nature。

setcitestyle, bibpunct, citestyle 的设置会覆盖宏包加载时的样式。

9. 可以使用如下命令进一步格式化:

```
\bibsection \bibpreamble \bibfont \citenumfont \bibnumfmt \bibhang \bibsep
```

10. 标注的排序和压缩, 使用 sort&compress 选项。

11. 顺序编码样式中的, 分组引用, 可以使用 merge 选项, 然后使用 \* 来构成分组。比如: `\citep{feynmann,*salam,*epr}` 会形成一个数字。

12. 作者年制中的长标注, 比如第一次引用时, 使用长标注。可以使用 `\citet*` 等命令。

13. 常用选项包括:

round (default) for round parentheses;  
square for square brackets;  
curly for curly braces;  
angle for angle brackets;  
semicolon (default) to separate multiple citations with semi-colons;  
colon the same as semicolon, an earlier mistake in terminology;  
comma to use commas as separators;  
authoryear (default) for authoryear citations;  
numbers for numerical citations;  
super for superscripted numerical citations, as in Nature;  
sort orders multiple citations into the sequence in which they appear in the list of references;  
sort&compress as sort but in addition multiple numerical citations are compressed if possible (as 36, 15);  
compress to compress without sorting, so compression only occurs when the given citations would produce an ascending sequence of numbers;  
longnamesfirst makes the first citation of any reference the equivalent of the starred variant (full author list) and subsequent citations normal (abbreviated list);  
sectionbib redefines thebibliography to issue section\* instead of chapter\*; valid only for classes with a chapter command; to be used with the chapterbib package;  
nonamebreak keeps all the authors' names in a citation on one line; causes overfull hboxes but helps with some hyperref problems;  
merge to allow the \* prefix to the citation key, and to merge such a citation's reference with that of the previous citation;  
elide to elide common elements of merged references, like the authors or year; mcite to recognize (and ignore) the merging syntax

14. 注意: 当出现 `unskip` 命令不能出现在竖直环境中的错误时, 说明引用命令没有进入水平模式, 只要在引用命令前加上文本即可。

## 4.5 chapterbib 宏包用法简介

### 4.5.1 基本概念

分章文献的基本原理是利用 `include`, 来对不同的章生成不同的 `aux` 文件, 然后对各个 `aux` 文件信息仅需解析, 然后处理 `bib` 文件, 进而生成多个 `bbl` 文件即多个 `thebibliography` 环境, 进而可以生成多个文献表。

因此对于利用 `bibtex` 处理 `bib` 文件的方法, 每个文件中必须要 `\bibliographystyle` 和 `\bibliography` 命令指出相应的 `bst` 样式和 `bib` 文件。才能让 `bibtex` 读取每个 `aux` 文件时获取足够的信息。

`chapterbib` 也提供了 `cbunit` 和 `\cbinput` 命令, 可以不基于 `include` 方法来生成多个文献表。但是这种方法需要比较麻烦的编译过程, 所以不推荐使用。

### 4.5.2 用法和注意事项

1. 在 book 和 report 类中, 参考文献的标题会被当做是一个 chapter\* 的标题, 但若使得参考文献的标题降一个层级, 比如降到 section\*。那么可以使用 sectionbib 选项, 但与 natbib 连用时, 该选项会失效, 替代的方法是给 natbib 设置这个选项。

2. 要在主文档中使用一个完全不同的参考文献, 那么不能再使用 \bibliography 命令, 因为主文档 bbl 会包含子文档的。所以只能使用手写的 thebibliography 环境。

3. 要在主文档生成一个包含所有子文档文献的文献表, 那么可以使用 \bibliography 命令。而要避免报错, \bibliographystyle 必须要放在所有的 include 之前。或者也可以在不同的编译步使用或不使用 [rootbib] 选项来实现。

4. 要把所有各章的文献表收集起来放到主文档中, 那么可以使用 [gather] 选项, 然后在主文档中使用 bibliography 命令。也可以通过定义 \StartFinalBibs 来重定义文献表的标题。

5. 如果文献表不仅要收集起来, 也需要在各章显示, 那么可以使用 [duplicate] 选项。

### 4.6 bibunits 宏包用法简介

bibunits 宏包提供了多参考文献表 (参考文献分节、局部参考文献表) 功能。参考文献分节可以是章 (chapter)、节 (section) 或 bibunit 环境。宏包自动的将每一分节的文献引用信息写入到独立的辅助文件 (aux) 中便于 bibtex 处理。同时也可以生成一个全局的参考文献表。

文献的分节通常使用 bibunit 环境, 也可以利用 \bibliographyunit[chapter] 或 \bibliographyunit[section] 使 chapter 或 section 变成为一个文献分节。在节内部引用文献, 并使用 \putbib[BibTeX files] 命令输出文献表。每个文献节可以使用指定的样式 (即给环境带上一个可选参数, 比如 \begin{bibunit}[plain] 或者使用默认的样式 (使用 \defaultbibliographystyle{style} 来指定)。文献数据库 (.BIB 文件) 除了使用 putbib 指定也可以使用默认的文件 (使用 \defaultbibliography{BibTeX files} 指定, 使用默认的文件时, putbib 命令无需再给出文件名参数)。

全局的文献可以利用 \cite\* 或 \nocite\* 命令来引用, 然后使用全局的 (即不在 bibunit 环境中的) \bibliography[BibTeX files] 和 \bibliographystyle[style] 来输出, 注意这两个命令需要放在 begin{document} 后面。一个示例如下:

示例 113 采用 bibunits 生成局部和全局参考文献示例

代码

```

1
2 \begin{filecontents*}{\jobname.bib}
3 @phdthesis{zhao2003,
4   author = {赵温波},
5   title = {径向基概率神经网络研究},
6   school = {中国科学技术大学},
7   year = {2003},
8   type = {[博士]},
9   address = {合肥},
10  pages={21-24}
11 }
12
13 @Inproceedings{Nemec1997-209-214,
14   Title = {Force control of redundant robots},
15   Author = {B Nemec},
16   Booktitle = {Processings of Symposium on Robot Control},
17   shortbooktitle = {(PSRC)},
18   Pages = {209-214},
19   Country = {Nantes France},

```

```
20   Year = {1997}
21 }
22
23 @Article{Chiani1998-2998-3008,
24   Title = {Error probability for block codes over channels with block interference},
25   Author = {Chiani, M.},
26   Journal = {IEEE Trans. Inf. Theory},
27   shortbooktitle = {(ITIT)},
28   Number = {7},
29   Pages = {2998-3008},
30   Volume = {44},
31   Year = {1998}
32 }
33 \end{filecontents*}
34
35 \documentclass{ctexart}
36 \usepackage{bibunits}
37 \usepackage{natbib}
38
39 \begin{document}
40 \bibliographystyle{plainnat}
41
42 \begin{bibunit}[plainnat]
43   \cite{zhao2003}
44   \putbib[\jobname]
45 \end{bibunit}
46
47 \begin{bibunit}[plainnat]
48   \citet{Nemec1997-209-214}
49   \putbib[\jobname]
50 \end{bibunit}
51
52 \begin{bibunit}[plainnat]
53   \citep{Chiani1998-2998-3008}
54   \putbib[\jobname]
55 \end{bibunit}
56
57 \cite*{zhao2003,Nemec1997-209-214,Chiani1998-2998-3008}
58 \bibliography{\jobname}
59
60 \end{document}
```

#### 4.7 multibib 宏包用法简介

multibib 包用于在一个文档中生产多个参考文献，这不同于在多个不同的子文档中生成多个参考文献（类似于 bibunits、chapterbib 宏包的基本功能）。

宏包使用 `\newcites` 来为每个文献表生成自定义的标注和文献表生成命令，这些命令的语法和意义与标准的 latex 命令类似，所以很容易使用。

每个文献表可以使用各自的样式和文献数据库。引用信息创建在单独的辅助文件中，用于 bibtex 程序读取。

该宏包与 cite, inlinebib, jurabib (to be loaded before multibib), natbib 等宏包兼容。

## 5 结论

通过本文的工作给出了 L<sup>A</sup>T<sub>E</sub>X 文档中文参考文献相关问题的 biblatex 解决方案, 包括: 基本参考文献生成、分章参考文献、指定格式的著录和标注样式等, 基本能够满足日常各类参考文献生成需求, 能够为 L<sup>A</sup>T<sub>E</sub>X 文档写作提供帮助。

## 6 后记

因水平有限, 难免存在一些错误, 请各位朋友批评指正。如果遇到什么问题, 也可以邮件联系, 欢迎交流。

### 参考文献

- [1] OETIKER T, PARTL H, HYNA I, et al. 一份不太简短的 L<sup>A</sup>T<sub>E</sub>X2 介绍[A/OL]. ChinaTEX 论坛, 译. 5.10. 2017 [2018-04-23]. <https://ctan.org/pkg/lshort-zh-cn> (引用页: 1).
- [2] 刘海洋. LaTeX 入门[M]. 北京: 电子工业出版社, 2013 (引用页: 1).
- [3] 胡伟. LaTeX2<sub>ε</sub> 完全学习手册[M]. 北京: 清华大学出版社, 2011 (引用页: 1).
- [4] DALY P W. Natural Sciences Citations and References (package natbib)[A/OL]. 8.31b. 2010 [2018-04-27]. <https://ctan.org/pkg/natbib> (引用页: 1).
- [5] HANSEN T. The multibib Package[EB/OL]. v1.4. (2008-12-10) [2018-05-04]. <https://www.ctan.org/pkg/multibib> (引用页: 1).
- [6] ARSENEAU D. Chapterbib multiple bibliographies in L<sup>A</sup>T<sub>E</sub>X[A/OL]. 2010 [2018-04-23]. <https://ctan.org/pkg/chapterbib> (引用页: 1).
- [7] SHELL M. Mciteplus: Enhanced Multicitations[A/OL]. 1.2. 2013 [2018-04-23]. <https://ctan.org/pkg/mciteplus> (引用页: 1).
- [8] SEINDAL R. Inline bibliography style[EB/OL]. (1994-12-06) [2018-05-04]. <https://www.ctan.org/pkg/inlinebib> (引用页: 1).
- [9] LEHMAN P, KIME P, BORUVKA A, et al. The Biblatex Package: Programmable Bibliographies and Citations[A]. 3.0. 2015 (引用页: 1).
- [10] 胡振震. 符合 GB7714-2015 标准的 biblatex 参考文献样式文件[A/OL]. 2016 [2018-04-23]. <https://ctan.org/pkg/biblatex-gb7714-2015> (引用页: 2, 6, 29).
- [11] BERRY K. TEX Live 指南—2016[A]. 2016 (引用页: 2, 29, 38).
- [12] 北京交通大学研究生公众号. 新学期定个小目标—学会在 LaTeX 中管理参考文献[EB/OL]. (2016-09-17). <http://www.latexstudio.net/archives/7488> (引用页: 11).
- [13] 如何在 LaTeX 写作中管理参考文献?[EB/OL]. (2016-08-12). <http://www.latexstudio.net/archives/7131> (引用页: 11).
- [14] 文献助手. 参考文献格式搜索神器上线了[EB/OL]. (2016-09-21). <http://www.latexstudio.net/archives/7560> (引用页: 11).
- [15] KOPKA H, DALY P W. A Guide to L<sup>A</sup>T<sub>E</sub>X and Electronic Publishing[M]. 4th ed. Harlow, England: Addison-Wesley, 2004 (引用页: 38).
- [16] BRAAMS J, CARLISLE D, JEFFREY A, et al. The L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> Sources[A]. 2015 (引用页: 38).
- [17] 求助 beamerposter 制作海报时参考文献样式的问题[EB/OL]. (2016). <http://bbs.ctex.org/forum.php?mod=viewthread&tid=153061&page=1&extra=#pid544787> (引用页: 46, 72).
- [18] TANTAU T, WRIGHT J, MILETIĆ V. The beamer class – User Guide for version 3.36[A/OL]. 2015. <http://bitbucket.org/rivanvx/beamer> (引用页: 73).
- [19] 胡海星. GBT7714-2005-BibTeX-Style[EB/OL]. (2012) [2020-05-17]. <https://github.com/Haixing-Hu/GBT7714-2005-BibTeX-Style> (引用页: 134).

- [20] LEE Z P. Gbt7714-bibtex-style[EB/OL]. (2016) [2020-05-17]. <https://github.com/CTeX-org/gbt7714-bibtex-style> (引用页: 134).



# 目 录

<b>0 引言</b>	<b>1</b>
<b>1 为什么使用 biblatex</b>	<b>2</b>
1.1 L <sup>A</sup> T <sub>E</sub> X 参考文献基本原理	2
1.2 BibLaTeX 参考文献生成的基本原理	3
1.3 biblatex 参考文献生成的优缺点	4
<b>2 参考文献常见需求与解决方案</b>	<b>5</b>
2.1 参考文献生成的常见需求	5
2.2 参考文献数据文件准备	6
2.2.1 手动文本文件生成	8
2.2.2 利用 winedt 生成	9
2.2.3 利用 texstudio 生成	9
2.2.4 利用 Jabref 软件生成	9
2.2.5 随 L <sup>A</sup> T <sub>E</sub> X 文档生成	11
2.2.6 获取标准 bib 文件	11
2.2.7 从 pdf 或文本文件转换 bib 文件	11
2.2.8 利用 biber 转换输出新的 bib 文件	12
2.2.9 利用 bibmap 程序修改 bib 文件	12
2.3 L <sup>A</sup> T <sub>E</sub> X 文档的组成	12
2.3.1 文档源文件基本结构	12
2.3.2 biblatex 宏包和样式的加载、指定参考文献数据源	13
2.3.3 文献的引用 (标注)	15
2.3.4 文献表的打印 (输出)	15
2.3.5 参考文献正反超链接	15
2.4 文档的编译	16
2.4.1 编译命令	16
2.4.2 利用 winedt 编译	17
2.4.3 利用 texstudio	17
2.4.4 命令行或脚本	19
2.4.5 常见的编译出错问题	19
2.5 分章参考文献和书后参考文献	21
2.5.1 利用 refsection 环境分章	21
2.5.2 利用宏包选项 refsection 分章	21

2.5.3	统一的全局参考文献	23
2.5.4	正文和附录区分的参考文献表	24
2.6	参考文献著录和标注样式	27
2.6.1	标准样式	27
2.6.2	其它常用样式	28
2.6.3	biblatex-gb7714-2015 样式包	29
2.6.4	标注命令	30
2.6.5	参考文献文境	31
2.7	参考文献标题格式	31
2.7.1	加入目录超链接	31
2.7.2	重定义 heading	32
2.7.3	利用 titlesec	36
2.8	参考文献表内容格式	37
2.8.1	一般设置方法	37
2.8.2	段落格式和局部字体修改	38
2.8.3	文献中的特殊字符	38
2.8.4	自定义环境举例	40
2.8.5	表格形式的参考文献表	42
2.8.6	参考文献表行溢出问题	43
2.8.7	参考文献表内断行和分页问题	44
2.8.8	英文文献中字符串的字母大小写问题	44
2.8.9	英文文献中期刊名的缩写问题	45
2.8.10	修改指定参数文献的格式	46
2.8.11	修改指定作者的格式	46
2.8.12	修改文献表中日期的格式	46
2.8.13	修改文献表中标点的全半角形式 (中英文形式)	46
2.9	专题: 参考文献格式的修改	46
2.9.1	通过宏包选项改变样式	47
2.9.1.1	作者输出数量	47
2.9.1.2	文献表排序	47
2.9.1.3	标注排序	47
2.9.1.4	文献数据项的输出控制	47
2.9.1.5	标注后面的标点控制	48
2.9.2	样式的临时定制	48
2.9.3	分隔符/定界符环境	50
2.9.4	标签的格式和装饰	50
2.9.5	label 类的域及其应用	51
2.10	专题: 筛选、缩略信息、排序、索引	51

2.10.1	文献分类筛选的不同方法	51
2.10.2	筛选打印示例	52
2.10.3	筛选的多个文献表的顺序编码标签	54
2.10.4	文献缩略信息打印	54
2.10.5	在标注和文献表中使用缩略信息	55
2.10.6	文献信息的动态/静态替换	57
2.10.7	文献表和标注的排序	58
2.10.8	排序模板和排序的手动调整	59
2.10.9	文献信息索引	60
2.11	专题: 多语言文献	60
2.11.1	多语言混合文献表	60
2.11.2	多语言对照的文献表	61
2.11.2.1	利用条目集类型满足双语文献要求	61
2.11.2.2	利用条目关联满足双语文献要求	63
2.11.3	关联与条目集的差异	64
2.12	专题: 脚注题注等环境中的引用和文献表	65
2.12.1	脚注中的引用	65
2.12.2	题注中的引用	66
2.12.3	小页环境中的引用	66
2.12.4	使用 biblatex 命令实现脚注文献表	67
2.12.5	使用 biblatex 命令和 footmisc 实现旁注文献表	68
2.12.6	使用自定义环境实现旁注文献表	69
2.13	专题: beamer 类中的参考文献	70
2.13.1	文末文献表和脚注文献表	71
2.13.2	文献表中的条目序号	72
2.13.3	beamer 中 biblatex 参考文献表内容格式	74
<b>3</b>	<b>参考文献样式定制方法</b>	<b>76</b>
3.1	gb7714-2015 样式设计与实现	76
3.1.1	动机	76
3.1.2	著录样式的设计途径	76
3.1.2.1	文献著录表的内容组织原理	77
3.1.3	兼容性问题的解决思路	78
3.1.3.1	biblatex 版本的兼容性	78
3.1.3.2	版本兼容的两个途径	82
3.1.3.3	GBK 编码文件的兼容性	85
3.1.4	中英文混合文献表的解决思路	86
3.1.4.1	利用字符流解析或者 xstring 宏包的函数	86
3.1.4.2	利用 CJKsymbol 进行的 cjk 判断	87

3.1.4.3	中文的本地化字符串	88
3.1.5	功能增加及格式还原的解决思路	89
3.1.5.1	biblatex 的宏包选项增加方法	89
3.1.5.2	作者缺省处理	91
3.1.5.3	文献表标签对齐	91
3.1.5.4	作者域格式	93
3.1.5.5	标题等的域格式	93
3.1.5.6	出版项缺省的格式	94
3.1.5.7	标点格式	95
3.1.5.8	文献类型和载体标识	95
3.1.5.9	域输出的限制	97
3.1.6	著录格式控制	97
3.1.6.1	文献表环境和打印输出	97
3.1.6.2	作者格式控制	97
3.1.6.3	超出数量的作者列表处理	100
3.1.6.4	日期格式控制	100
3.1.6.5	条目标签中的日期格式	101
3.1.6.6	版本项格式控制	102
3.1.6.7	卷期域格式控制	102
3.1.6.8	标点和空格的特点和机制	103
3.1.7	动态数据修改	105
3.1.7.1	基本原理和方法	105
3.1.7.2	数据模型	106
3.1.7.3	用于关联条目的动态修改的命令封装及关联条目格式	108
3.1.8	标注样式的设计途径	109
3.1.8.1	标签中的标点和间隔符号	109
3.1.8.2	标注 (引用) 标签中的作者数量	110
3.2	国标文档的完整复现	110
3.3	一个项目示例	111
3.3.1	介绍	111
3.3.2	bbx 文件开发	112
3.3.2.1	文献表环境	112
3.3.2.2	作者域处理	113
3.3.2.3	条目类型及其使用的域	115
3.3.2.4	article 类型处理	116
3.3.2.5	inproceedings 类型处理	117
3.3.2.6	book 类型处理	119
3.3.2.7	inbook 类型处理	119

3.3.2.8	report 类型处理	120
3.3.2.9	thesis 类型处理	121
3.3.2.10	patent 类型处理	122
3.3.2.11	standard 类型处理	122
3.3.2.12	online 类型处理	123
3.3.2.13	全局选项及 texlive 兼容性处理	123
3.3.3	cbx 文件开发	124
3.3.4	小结	125
3.4	样式定制中的一些技巧	125
3.4.1	选项的范围限制	125
3.4.2	排序格式的宏重用	126
3.4.3	不同位置/方式的代码处理	127
3.4.4	数据模型定义	128
3.4.5	日期格式控制选项定义	129
3.4.6	美式标点	131
3.4.7	不同条目类型的域格式	131
3.4.8	文献表中相同作者用 dash 代替	132
4	参考文献的传统解决方案	132
4.1	传统解决方案的基本原理	132
4.2	基于 thebibliography 环境的方法	133
4.3	基于 bibtex 的方法	133
4.3.1	基本操作	133
4.3.2	文档编译	133
4.3.3	常用著录样式	134
4.3.4	常用标注 (引用) 样式	134
4.3.5	注意事项	134
4.4	natbib 宏包用法简介	135
4.4.1	基本概念	135
4.4.2	用法和注意事项	135
4.5	chapterbib 宏包用法简介	136
4.5.1	基本概念	136
4.5.2	用法和注意事项	137
4.6	bibunits 宏包用法简介	137
4.7	multibib 宏包用法简介	138
5	结论	139
6	后记	139

目录	141
插图	146
表格	147
示例	147

## 插 图

1	文本编辑器生成 bib 文件	8
2	winedt 生成 bib 文件	9
3	texstudio 生成 bib 文件	9
4	Jabref 软件生成 bib 文件	10
5	Jabref 软件增加参考文献条目	10
6	Jabref 软件录入参考文献信息	10
7	最小工作示例的结果	13
8	texstudio 设置默认的编译工具	18
9	texstudio 配置构建并查看命令	18
10	linux 下 texstudio 设置命令路径	18
11	分章参考文献举例-序章	22
12	分章参考文献举例-章一	22
13	设置宏包选项实现分章参考文献	23
14	全局参考文献综合示例	24
15	利用 titlesec 局部修改参考文献标题格式	37
16	参考文献表格式默认字体字号六	39
17	参考文献表格式仿宋字体字号六	39
18	参考文献表格式等宽字体字号六	39
19	参考文献表格式楷书字体字号六	39
20	以当前条目为基准对齐的参考文献表环境	42
21	参考文献样式快速定制	49
22	国标风格的参考文献样式快速定制 A	49
23	文献表筛选打印	53
24	论文集信息打印	55
25	标题和期刊信息打印	55
26	不同缩略信息在一个表中打印	56
27	不同缩略信息在一个表中打印	56
28	多语言混合文献表	60
29	双语文献参考文献	62
30	脚注中引用参考文献	65

31	题注中引用参考文献	66
32	小页环境中引用参考文献	67
33	脚注中的参考文献表	68
34	旁注中的参考文献表	69
35	脚注和旁注中参考文献表	71
36	beamer 类脚注中的参考文献表	72
37	beamer 类全局参考文献表	73
38	一个自定义 block 中参考文献表 (序号丢失)	73
39	beamer 中 biblatex 使用 gb7714-2015 定义的 align=gb7714-2015 选项后的文献表	74
40	beamer 中 biblatex 利用内部主题设置文献表标签后的文献表	74
41	自定义 block 中使用 gb7714-2015 定义的 align=gb7714-2015 选项后的文献表	74
42	beamer 中 biblatex 利用颜色主题设置文献颜色	75
43	定制项目文献表示例	112
44	定制项目引用标签示例	112
45	样式定制结果	126

## 表 格

1	BibTeX 格式的条目类型和域	7
2	BibTeX 格式中的域说明	8
3	常用的 biblatex 样式	28
4	支持的条目类型	115

## 示 例

1	典型的 thebibliography 环境	2
2	参考文献信息基本格式即 BibTeX 格式	6
3	利用 filecontents 环境随 latex 文档生成 bib	11
4	利用 gb7714texttobib.pl 将参考文献文本转换为 bib 文件	12
5	利用 biber 转换出新的 bib 文件	12
6	文档源文件的代码基本结构	12
7	biblatex 参考文献生成最小工作示例	13
8	biblatex 宏包、数据源和样式的加载	13
9	添加参考文献数据源的两种方式	14
10	参考文献的引用命令	15
11	不标注的文献的引用命令	15
12	参考文献表打印命令	15



13	参考文献反向超链接选项	15
14	xelatex 编译命令	16
15	pdflatex 编译命令	16
16	latexmk 编译命令	16
17	texstudio 一键编译过程提示	17
18	命令行手动输入的命令	19
19	window 下的 bat 脚本文件	19
20	linux 下的 sh 脚本文件	19
21	分章参考文献	21
22	利用宏包选项实现分章参考文献	22
23	利用 refsection 的全局参考文献	23
24	全局参考文献综合示例	24
25	区别于正文的附录带前缀参考文献示例	25
26	各附录不同前缀的附录单文献表示例	26
27	natbib 和 mciteplus 等宏包提供的标注名	30
28	常见的标注命令	30
29	手动加入目录链接	31
30	使用 bibintoc 加入目录链接	31
31	biblatex 对于目录的影响	32
32	重设 heading 选项参数	32
33	文档类提供 ctex 修改的页眉页脚举例	32
34	利用 fancyhdr 生成页眉页脚举例	33
35	利用 titleps 生成页眉页脚举例	34
36	利用 titlesec 局部修改参考文献标题格式	36
37	参考文献表内容的格式	37
38	参考文献表内容的字体颜色格式	38
39	参考文献表内容格式修改举例	38
40	文献中的特殊字符直接显示方法	40
41	自定义参考文献表环境举例	40
42	基于 list 的参考文献表标签的左对齐	41
43	以当前条目为基准对齐的参考文献表环境	41
44	一个简单的表格式参考文献表设置	43

45	参考文献表行溢出问题解决	43
46	参考文献表 url 导致行溢出问题解决	44
47	一个参考文献条目段落内部的断行	44
48	利用 shortbooktitle 域记录缩写通过样式调整添加	45
49	利用动态数据修改来去除数据域	47
50	利用钩子在数据应用前去除数据域	47
51	autopunct 选项与脚注之后句点处理	48
52	参考文献样式定制举例	48
53	参考文献样式定制举例-国标样式	49
54	标签的格式控制	50
55	标签的包围符号全局修改	50
56	标签的包围符号局部修改	50
57	分类筛选打印参考文献表	52
58	category 和 check 筛选	52
59	多个筛选文献表的顺序编码	54
60	论文集信息打印	54
61	标题和期刊信息打印	54
62	不同缩略信息在一个表中打印	55
63	域信息替换方法	56
64	在标注和著录表中使用缩略信息	56
65	string 方法继承数据	57
66	sourcemap 处理域信息	58
67	中文文献排序时的 biber 选项	59
68	文献信息索引	60
69	设置 set 条目集用于双语文献动态方法	61
70	设置 set 条目集用于双语文献动态方法	61
71	设置 set 条目集用于双语文献动态方法	62
72	设置 set 条目集用于双语文献静态方法	62
73	在 bib 文件中正确设置 set 条目集的静态方法	63
74	在 bib 文件中正确设置关联条目的静态方法	63
75	设置关联条目的动态方法	64
76	设置 set 动态条目集的 mcite 方法	64

77	脚注中引用参考文献	65
78	题注中引用参考文献	66
79	小页环境中引用参考文献	67
80	脚注参考文献表	67
81	旁注参考文献表	68
82	另一种脚注和旁注参考文献表	70
83	beamer 中使用 biblatex 参考文献	71
84	beamer 中 biblatex 使用 gb7714-2015 定义的 align=gb7714-2015 选项	73
85	beamer 中 biblatex 利用内部主题设置文献表标签	73
86	beamer 中 biblatex 利用颜色主题设置文献颜色	75
87	texlive2016 中 biblatex3.4 版 Name 域格式输入参数的修改	97
88	重定义参考文献环境确定缩进	112
89	作者域处理	113
90	本地化字符串处理	114
91	各条目使用的域	115
92	article 条目处理	116
93	inproceedings 条目处理	118
94	book 条目处理	119
95	inbook 条目处理	119
96	report 条目处理	121
97	standard 条目处理	122
98	online 条目处理	123
99	texlive 兼容性处理	123
100	cbx 设置宏包选项	124
101	cbx 重定义实现要求	124
102	针对条目类型的选项设置	125
103	排序格式中的宏重用	126
104	不同位置的代码处理	127
105	引用位置的本地化字符串/空格处理	127
106	数据模型定义	128
107	日期相关选项的定义	129
108	日期选项的增加	130

---

109	美式标点 . . . . .	131
110	不同条目类型的域格式 . . . . .	131
111	相同作者用 dash 代替 . . . . .	132
112	采用 bibtex 的文献生成示例 . . . . .	133
113	采用 bibunits 生成局部和全局参考文献示例 . . . . .	137